

University of South Wales



2064758

Printed & Bound by

Abbey Bookbinding



Unit 3 Clos Menter
Gabalfa Workshops
Western Avenue
Cardiff CF14 3AY

T: +44 (0) 29 2062 3290
F: +44 (0) 29 2062 5420
E: info@bookbindersuk.com
W: www.bookbindersuk.com

Viability of High Availability V band Satellite Communication using Hybrid Fade Mitigation

By Andrew Chambers

University of Glamorgan

A thesis submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

April 2008



Certificate of Research

This is to certify that, except where specific reference is made, the work described in this thesis is the result of the candidate's research. Neither this thesis, nor any part of it, has been presented, or is currently submitted, in candidature for any degree at any other University.

Signed

A. Charles
.....
Candidate

Date

27th June 2008
.....

Signed

[Signature]
.....
Director of Studies

Date

27 June 2008
.....

Abstract

The C and Ku bands that have been the backbone of satellite telephony and data networks in the past are limited in the amount of bandwidth they provide. Consequently, some broadband Internet services have started to move into the Ka band in order to utilise the bandwidth at these higher frequencies. The use of higher frequencies led to fade mitigation design considerations that had not previously been necessary, as fixed power margins had been sufficient to guarantee high availability.

This thesis reviews the majority of fade mitigation techniques that are available and simulates a scenario with several combined in a hybrid scheme. The simulations were made possible by several innovative components, including a near-optimum short-term fade predictor, a procedure capable of simulating the effect of site diversity with a single time series of attenuation and a dual-channel protocol structure that is able to increase availability of control information at V band.

The near-optimum short-term fade predictor is compared with several other short-term fade predictors and is shown to perform better in terms of the root-mean-square error and by the cumulative distribution of over-predictions. Also noteworthy is the fact that predictors based on fade slope perform very poorly when compared with signal processing and statistical methods of prediction.

The procedure capable of producing the effects of site diversity on a single time series of attenuation is described. The results of this procedure are compared with the ITU-R model for site diversity improvement and an acceptable fit is shown.

A dual-channel protocol structure capable of increasing control information availability at V band is presented and the availability of this scheme is compared to the availability of current fixed-margin C and Ku band satellite links. While the availability at V band improves via this technique, it does not improve enough to compare favourably with that found in the lower frequency bands.

A software turbo decoder capable of decoding the DVB-RCS turbo code is also described in this work since this was not available through other means. The fade dynamics at V band are reviewed and compared with the ITU-R models. The fade slope model gives a good visual fit but the fade duration model is very inaccurate.

Finally, the simulation results show that high availability satellite communication is possible at V band, critically, if both site diversity and frequency diversity are available to mitigate the hub and VSAT fading respectively. Signal processing and power control are also found to be complementary techniques.

Table of Contents

Abstract	i
Table of Contents	ii
List of Tables.....	iv
List of Figures	v
Abbreviations and Acronyms.....	ix
Acknowledgements	xii
Chapter 1 - Introduction	1
1.1 Introduction	2
1.2 Objectives of Research.....	5
1.3 Overview – Content of Thesis.....	8
1.4 Summary of Novel Contribution.....	11
Chapter 2 – Fade Mitigation and Control	12
2.1 Introduction	13
2.2 Fade Mitigation	14
2.2.1 Introduction	14
2.2.2 Signal Processing	15
2.2.3 Adaptive Resource Sharing.....	21
2.2.4 Diversity	23
2.2.5 Power Control	29
2.2.6 FMT Scenario Overview.....	31
2.3 Fade Control.....	33
2.3.1 Introduction	33
2.3.2 A Generic Fade Controller	33
2.3.3 Fade Detection	35
2.3.4 Short-term Fade Prediction	37
2.3.5 Resource Allocation	37
2.3.6 Protocol Design	38
Chapter 3 – Second Order Fade Dynamics at V Band	39
3.1 Introduction	40
3.2 Fade Slope	41
3.3 Fade Duration.....	53
3.4 Inter-fade Duration.....	61
3.5 Frequency Scaling.....	63
3.6 Summary	63
Chapter 4 – Duo-binary Turbo Coding and Decoding.....	65
4.1 Introduction	66
4.2 Turbo Encoding.....	67
4.2.1 The Encoding Process	67
4.2.2 The Encoder	68
4.2.3 Interleaving	70
4.2.4 Puncturing	71
4.3 Turbo Decoding	72
4.3.1 Overview	72
4.3.2 Trellis Representation	73

4.3.3 Turbo Decoding Process	74
4.4 Software Implementation	82
4.5 Performance	85
4.6 Conclusions	90
Chapter 5 – Site Diversity Approximation	91
5.1 Introduction	92
5.2 Dual-site Concurrent Rain Rate Analysis	93
5.3 Results	100
5.4 Conclusions	101
Chapter 6 – Short-term Fade Prediction	103
6.1 Introduction	104
6.2 Review of Short-term Predictors	105
6.2.1 Best-fit Slope Prediction	105
6.2.2 Mean Slope Prediction	105
6.2.3 ARMA Prediction	105
6.2.4 Statistical Prediction	109
6.3 An ADALINE Predictor	110
6.4 A Hybrid Signal Processing Predictor	119
6.5 Short-term Prediction Control	126
6.6 Conclusions	128
Chapter 7 – Preliminary System Design and Pre-processing	130
7.1 Introduction	131
7.2 Data Pre-processing	132
7.3 System Simulation Scenario	140
7.4 A Dual-channel Protocol for Bi-directional Satellite Internet with FMTs ...	144
7.4.1 Protocol Structure	144
7.4.2 Protocol Functionality	148
7.4.3 Control Channel Availability	151
7.5 Conclusions	155
Chapter 8 – Preliminary Investigation of Combined FMT Performance	156
8.1 Introduction	157
8.2 Design Considerations	157
8.3 Signal Processing Scenario	163
8.4 Frequency Diversity Scenario	177
8.5 Hybrid FMT Scenario	186
8.6 Conclusions	196
Chapter 9 – Final Design, Simulation and Results	198
9.1 Introduction	199
9.2 Design Considerations	199
9.3 Final Design and Results	203
9.3.1 Software Approach	203
9.3.2 Design and Results of a Four-station Frequency Diversity Approach	206
9.3.3 Design and Results of a Single-station Frequency Diversity Approach	220
9.4 Conclusions	232
Chapter 10 – Conclusions and Further Work	233
10.1 Conclusions	234
10.2 Further Work	240
References	241
Appendices	251

List of Tables

TABLE 1.1: FREQUENCIES AND ASSOCIATED BANDS UNDER CONSIDERATION	4
TABLE 2.1: SUMMARY OF UTILISED FMT TECHNIQUES	32
TABLE 4.1: CIRCULATION STATE CORRESPONDENCE TABLE	69
TABLE 4.2: FIXED INTERLEAVING PERMUTATION PARAMETERS.....	70
TABLE 4.3: CALCULATION OF PERMUTATION PARAMETER P	71
TABLE 4.4: PUNCTURING PATTERN FOR THE DVB-RCS TURBO CODE	71
TABLE 5.1: JOINT STATISTICS OF RAIN RATE AT THE BFRM GAUGE (ROWS) AND THE GOOD GAUGE (COLUMNS).....	96
TABLE 5.2: RATIO OF CONCURRENT LIGHT RAIN TO TOTAL LIGHT RAIN AND CONCURRENT HEAVY RAIN TO TOTAL HEAVY RAIN FOR THE 40 GHz ATTENUATION TIME SERIES	98
TABLE 5.3: RATIO OF CONCURRENT LIGHT RAIN TO TOTAL LIGHT RAIN AND CONCURRENT HEAVY RAIN TO TOTAL HEAVY RAIN FOR THE 50 GHz ATTENUATION TIME SERIES	98
TABLE 7.1: DATA FLAG INTERPRETATION.....	133
TABLE 7.2: INTERNET USAGE STATISTICS	142
TABLE 7.3: CONSTANT PARAMETER VALUES FOR VSAT NETWORK	152
TABLE 8.1: FMT ADAPTIVE MODULATION AND CODING SCHEME	163
TABLE 8.2: MOST SIGNIFICANT LINK BUDGET PARAMETERS FOR THE SIGNAL PROCESSING FORWARD LINK	165
TABLE 8.3: MOST SIGNIFICANT LINK BUDGET PARAMETERS FOR THE SIGNAL PROCESSING RETURN LINK	166
TABLE 8.4: MOST SIGNIFICANT LINK BUDGET PARAMETERS FOR THE FREQUENCY DIVERSITY FORWARD LINK	178
TABLE 8.5: MOST SIGNIFICANT LINK BUDGET PARAMETERS FOR THE FREQUENCY DIVERSITY RETURN LINK	179
TABLE 8.6: MOST SIGNIFICANT LINK BUDGET PARAMETERS FOR THE HYBRID FMT FORWARD LINK	187
TABLE 8.7: MOST SIGNIFICANT LINK BUDGET PARAMETERS FOR THE HYBRID FMT RETURN LINK	188
TABLE 9.1: FMT CONFIGURATIONS	207
TABLE 9.2: MOST SIGNIFICANT LINK BUDGET PARAMETERS FOR THE FORWARD LINK WITH FOUR STATIONS AT <i>KU</i> BAND	208
TABLE 9.3: MOST SIGNIFICANT LINK BUDGET PARAMETERS FOR THE RETURN LINK WITH FOUR STATIONS AT <i>KU</i> BAND	209
TABLE 9.4: FMT CONFIGURATIONS	220
TABLE 9.5: MOST SIGNIFICANT LINK BUDGET PARAMETERS FOR THE FORWARD LINK WITH ONE STATION AT <i>KU</i> BAND	221
TABLE 9.6: MOST SIGNIFICANT LINK BUDGET PARAMETERS FOR THE RETURN LINK WITH ONE STATION AT <i>KU</i> BAND	222

List of Figures

FIGURE 1.1: RAIN ATTENUATION NOT EXCEEDED FOR A SPECIFIED AVAILABILITY AS A FUNCTION OF FREQUENCY.....	4
FIGURE 1.2: SYSTEM ARCHITECTURE	5
FIGURE 2.1: BER PERFORMANCE OF UNCODED PSK SCHEMES	16
FIGURE 2.2: FEC CODING GAIN.....	18
FIGURE 2.3: SITE DIVERSITY ILLUSTRATION	24
FIGURE 2.4: CDF OF ANNUAL RAIN ATTENUATION EXCEEDED FOR A SINGLE AND DUAL SITE SCENARIO.	25
FIGURE 2.5: ORBITAL DIVERSITY ILLUSTRATION	27
FIGURE 2.6: CDFs OF FREQUENCY DIVERSITY SHOWING A) DOWNLINK DIVERSITY GAIN AND B) UPLINK DIVERSITY GAIN FOR A V BAND COMMUNICATIONS SYSTEM WITH A KU BAND SHARED RESOURCE...	28
FIGURE 2.7: FMTs WITH RESPECT TO SYSTEM ARCHITECTURE	32
FIGURE 2.8: A GENERIC FADE CONTROLLER	35
FIGURE 3.1: FADE SLOPE AND FADE DURATION	40
FIGURE 3.2: FREQUENCY SCALING OF ATTENUATION OF VARIOUS FREQUENCIES FROM 7 GHz	43
FIGURE 3.3: NUMBER OF SAMPLES USED TO GENERATE FADE SLOPE STATISTICS FROM 1 TO 20 dB AT 20, 40 AND 50 GHz.....	45
FIGURE 3.4: STANDARD DEVIATION OF FADE SLOPE	46
FIGURE 3.5: S PARAMETER CALCULATED FOR EACH ATTENUATION THRESHOLD AND THE ZERO-OFFSET BEST FIT.	47
FIGURE 3.6: PDF OF FADE SLOPE FOR 50 GHz TIME SERIES	49
FIGURE 3.7: CDF OF FADE SLOPE FOR 50 GHz TIME SERIES	49
FIGURE 3.8: CDF OF FADE RISE RATE AND FADE FALL RATE AT VARIOUS LEVELS OF ATTENUATION	51
FIGURE 3.9: DIURNAL VARIATION OF FADE SLOPE STANDARD DEVIATION AND ATTENUATION AT 50 GHz	52
FIGURE 3.10: SEASONAL VARIATION OF FADE SLOPE STANDARD DEVIATION AND ATTENUATION AT 50 GHz	53
FIGURE 3.11: CDF OF INVALID DATA DURATIONS	55
FIGURE 3.12: CDF SHOWING THE PROBABILITY THAT A GIVEN DURATION IS EXCEEDED AT THE 1 dB THRESHOLD FOR THE 50 GHz TIME SERIES	57
FIGURE 3.13: NUMBER OF EVENTS EXCEEDING ABSCISSA FOR THE 50 GHz TIME SERIES	58
FIGURE 3.14: CDF SHOWING THE PROBABILITY OF TOTAL FADE TIME EXCEEDING ABSCISSA AT THE 1 dB THRESHOLD FOR THE 50 GHz TIME SERIES	59
FIGURE 3.15: TOTAL DURATION OF EVENTS EXCEEDING ABSCISSA FOR THE 50 GHz TIME SERIES.....	59
FIGURE 3.16: FADE DURATION AS A FUNCTION OF ATTENUATION FOR VARIOUS EXCEEDANCE PROBABILITIES FOR THE 50 GHz TIME SERIES	60
FIGURE 3.17: NUMBER OF INTER-FADES EXCEEDING ABSCISSA AT 50 GHz	61
FIGURE 4.1: BLOCK DIAGRAM OF DVB-RCS ENCODING PROCESS	67
FIGURE 4.2: COMPONENT ENCODER FOR DVB-RCS TURBO CODE.....	68
FIGURE 4.3: CODEWORD ORDERING.....	72
FIGURE 4.4: TRELLIS ASSOCIATED WITH DVB-RCS CONSTITUENT ENCODER/DECODER	74
FIGURE 4.5: A DECODER FOR THE DVB-RCS TURBO CODE.....	75
FIGURE 4.6: CYLINDRICAL TRELLIS FOR TAIL-BITING DVB-RCS TURBO CODE	79
FIGURE 4.7: SOFTWARE FLOW DIAGRAM OF TURBO-DECODING PROCESS.....	83
FIGURE 4.8: BIT ERROR RATES FOR DVB-RCS TURBO CODE FOR ALL AVAILABLE CODE RATES WITH N=212 AND 8 DECODING ITERATIONS	86
FIGURE 4.9: FRAME ERROR RATES FOR DVB-RCS TURBO CODE FOR ALL AVAILABLE CODE RATES WITH N=212 AND 8 DECODING ITERATIONS	86
FIGURE 4.10: BIT ERROR RATES FOR VARIOUS BLOCK SIZES AT CODE RATE $\frac{1}{2}$ AND WITH 8 DECODING ITERATIONS	87
FIGURE 4.11: FRAME ERROR RATES FOR VARIOUS BLOCK SIZES AT CODE RATE $\frac{1}{2}$ AND WITH 8 DECODING ITERATIONS	88
FIGURE 4.12: BIT ERROR RATE COMPARISON FOR $\frac{1}{2}$ RATE DVB-RCS TURBO CODE AND $\frac{1}{2}$ RATE CONVOLUTIONAL CODE WITH CONSTRAINT LENGTH 7 AND SOFT VITERBI DECODING	89

FIGURE 4.13: FRAME ERROR RATE COMPARISON FOR $\frac{1}{2}$ RATE DVB-RCS TURBO CODE AND $\frac{1}{2}$ RATE CONVOLUTIONAL CODE WITH CONSTRAINT LENGTH 7 AND SOFT VITERBI DECODING	89
FIGURE 5.1: RAIN ATTENUATION CDF AT 40 GHZ	100
FIGURE 5.2: RAIN ATTENUATION CDF AT 50 GHZ	101
FIGURE 6.1: ADALINE FADE PREDICTOR.....	110
FIGURE 6.2: A RAIN EVENT ON THE 5 TH NOVEMBER 1997 SHOWING A) EVENT ATTENUATION AND B) PREDICTION ERROR FOR THE ADALINE PREDICTOR.....	114
FIGURE 6.3: ROOT MEAN SQUARE ERROR FOR EACH PREDICTOR FOR VARIOUS PREDICTION INTERVALS AT 20 GHZ	115
FIGURE 6.4: ROOT MEAN SQUARE ERROR FOR EACH PREDICTOR FOR VARIOUS PREDICTION INTERVALS AT 40 GHZ	115
FIGURE 6.5: ROOT MEAN SQUARE ERROR FOR EACH PREDICTOR FOR VARIOUS PREDICTION INTERVALS AT 50 GHZ	116
FIGURE 6.6: CDF OF UNDER-PREDICTIONS AT 20 GHZ FOR PREDICTION INTERVALS OF A) $K = 1$ AND B) $K = 10$	117
FIGURE 6.7: CDF OF UNDER-PREDICTIONS AT 40 GHZ FOR PREDICTION INTERVALS OF A) $K = 1$ AND B) $K = 10$	118
FIGURE 6.8: CDF OF UNDER-PREDICTIONS AT 50 GHZ FOR PREDICTION INTERVALS OF A) $K = 1$ AND B) $K = 10$	118
FIGURE 6.9: STRATIFORM EVENT AT 50 GHZ ON THE 5 TH NOVEMBER 1997, SHOWING A) EVENT ATTENUATION, B) ADALINE PREDICTOR ERROR, C) ARMA PREDICTOR ERROR AND D) THE PREDICTOR CHOICE.....	120
FIGURE 6.10: CONVECTIVE EVENT AT 50 GHZ ON THE 18 TH JANUARY 1998, SHOWING A) EVENT ATTENUATION, B) ADALINE PREDICTOR ERROR, C) ARMA PREDICTOR ERROR AND D) THE PREDICTOR CHOICE.....	121
FIGURE 6.11: ROOT MEAN SQUARE ERROR OF HYBRID PREDICTOR FOR VARIOUS PREDICTION INTERVALS AT 20 GHZ	122
FIGURE 6.12: ROOT MEAN SQUARE ERROR OF HYBRID PREDICTOR FOR VARIOUS PREDICTION INTERVALS AT 40 GHZ	123
FIGURE 6.13: ROOT MEAN SQUARE ERROR OF HYBRID PREDICTOR FOR VARIOUS PREDICTION INTERVALS AT 50 GHZ	123
FIGURE 6.14: CDF OF UNDER-PREDICTIONS FOR THE HYBRID PREDICTOR AT 20 GHZ FOR PREDICTION INTERVALS OF A) $K = 1$ AND B) $K = 10$	124
FIGURE 6.15: CDF OF UNDER-PREDICTIONS FOR THE HYBRID PREDICTOR AT 40 GHZ FOR PREDICTION INTERVALS OF A) $K = 1$ AND B) $K = 10$	125
FIGURE 6.16: CDF OF UNDER-PREDICTIONS FOR THE HYBRID PREDICTOR AT 50 GHZ FOR PREDICTION INTERVALS OF A) $K = 1$ AND B) $K = 10$	125
FIGURE 6.17: AN EVENT OCCURRING ON THE 12 TH JULY 2000 SHOWING A) THE ATTENUATION AND QoS THRESHOLD WHERE CONSECUTIVE PREDICTIONS, $K = 1$ AND B) THE ATTENUATION AND QoS THRESHOLD WHERE CONSECUTIVE PREDICTIONS, $K = 10$	127
FIGURE 6.18: PERFORMANCE OF SHORT-TERM PREDICTION CONTROL SHOWING A) THE PERCENTAGE OF SAMPLES FOR WHICH THE QoS THRESHOLD IS CROSSED AND B) THE PERCENTAGE OF SAMPLES FOR WHICH A CHANGE IN FMT PARAMETERS OCCURS	128
FIGURE 7.1: FLOW DIAGRAM FOR SOFTWARE THAT SELECTS WHICH DAYS ARE USEFUL	134
FIGURE 7.2: DATA ANOMALY OF THE FIRST TYPE AT 20 GHZ	135
FIGURE 7.3: DATA ANOMALY OF THE SECOND TYPE AT 40 GHZ	136
FIGURE 7.4: DATA ANOMALY OF THE THIRD TYPE AT 20 GHZ	136
FIGURE 7.5: FLOW DIAGRAM FOR SOFTWARE THAT SCALES OR INTERPOLATES BAD DATA SECTIONS	139
FIGURE 7.6: PDF OF TRAFFIC DISTRIBUTION FOR 620 STATIONS	143
FIGURE 8.1: FLOW DIAGRAM OF MATLAB PROGRAM OF SIGNAL PROCESSING FMT WITH SITE DIVERSITY	167
FIGURE 8.2: NUMBER OF RESOURCE REQUESTS PER STATION FOR SIGNAL PROCESSING SCHEME WITH SITE DIVERSITY	168
FIGURE 8.3: NUMBER OF SECONDS IN OUTAGE PER STATION FOR SIGNAL PROCESSING SCHEME WITH SITE DIVERSITY	169
FIGURE 8.4: PDF OF PERCENTAGE OF MF-TDMA FRAME DURATION REQUIRED FOR ALL STATIONS TO TRANSMIT AT THE CHOSEN BER ON THE FORWARD LINK WITH SITE DIVERSITY	170
FIGURE 8.5: CDF OF PERCENTAGE OF MF-TDMA FRAME REQUIRED FOR ALL STATIONS TO TRANSMIT AT THE DESIRED BER ON THE FORWARD LINK WITH SITE DIVERSITY	171

FIGURE 8.6: PDF OF PERCENTAGE OF MF-TDMA FRAME DURATION REQUIRED FOR ALL STATIONS TO TRANSMIT AT THE CHOSEN BER ON THE RETURN LINK WITH SITE DIVERSITY	172
FIGURE 8.7: CDF OF PERCENTAGE OF MF-TDMA FRAME REQUIRED FOR ALL STATIONS TO TRANSMIT AT THE DESIRED BER ON THE RETURN LINK WITH SITE DIVERSITY.....	173
FIGURE 8.8: NUMBER OF RESOURCE REQUESTS PER STATION FOR SIGNAL PROCESSING SCHEME WITHOUT SITE DIVERSITY.....	174
FIGURE 8.9: NUMBER OF SECONDS IN OUTAGE PER STATION FOR SIGNAL PROCESSING SCHEME WITHOUT SITE DIVERSITY.....	175
FIGURE 8.10: CDF OF PERCENTAGE OF MF-TDMA FRAME REQUIRED FOR ALL STATIONS TO TRANSMIT AT THE DESIRED BER ON THE FORWARD LINK WITHOUT SITE DIVERSITY	175
FIGURE 8.11: CDF OF PERCENTAGE OF MF-TDMA FRAME REQUIRED FOR ALL STATIONS TO TRANSMIT AT THE DESIRED BER ON THE RETURN LINK WITHOUT SITE DIVERSITY	176
FIGURE 8.12: FLOW DIAGRAM OF MATLAB PROGRAM OF FREQUENCY DIVERSITY FMT WITH SITE DIVERSITY	180
FIGURE 8.13: NUMBER OF RESOURCE REQUESTS PER STATION FOR FREQUENCY DIVERSITY SCHEME WITH SITE DIVERSITY.....	182
FIGURE 8.14: NUMBER OF SECONDS IN OUTAGE PER STATION FOR FREQUENCY DIVERSITY SCHEME WITH SITE DIVERSITY.....	183
FIGURE 8.15: PDF OF PERCENTAGE OF KU BAND RESOURCE REQUIRED FOR ALL STATIONS TO TRANSMIT AT THE CHOSEN BER WITH SITE DIVERSITY	184
FIGURE 8.16: CDF OF PERCENTAGE OF KU BAND RESOURCE REQUIRED FOR ALL STATIONS TO TRANSMIT AT THE DESIRED BER WITH SITE DIVERSITY	185
FIGURE 8.17: FLOW DIAGRAM OF MATLAB PROGRAM OF HYBRID FMT WITH SITE DIVERSITY	189
FIGURE 8.18: NUMBER OF RESOURCE REQUESTS PER STATION FOR HYBRID FMT SCHEME WITH SITE DIVERSITY	190
FIGURE 8.19: NUMBER OF SECONDS IN OUTAGE PER STATION FOR HYBRID FMT SCHEME WITH SITE DIVERSITY	191
FIGURE 8.20: PDF OF PERCENTAGE OF MF-TDMA FRAME DURATION REQUIRED FOR ALL STATIONS TO TRANSMIT AT THE CHOSEN BER ON THE FORWARD LINK OF THE HYBRID FMT SCENARIO WITH SITE DIVERSITY	192
FIGURE 8.21: CDF OF PERCENTAGE OF MF-TDMA FRAME REQUIRED FOR ALL STATIONS TO TRANSMIT AT THE DESIRED BER ON THE FORWARD LINK OF THE HYBRID FMT SCENARIO WITH SITE DIVERSITY ..	193
FIGURE 8.22: PDF OF PERCENTAGE OF MF-TDMA FRAME DURATION REQUIRED FOR ALL STATIONS TO TRANSMIT AT THE CHOSEN BER ON THE RETURN LINK OF THE HYBRID FMT SCENARIO WITH SITE DIVERSITY	194
FIGURE 8.23: CDF OF PERCENTAGE OF MF-TDMA FRAME REQUIRED FOR ALL STATIONS TO TRANSMIT AT THE DESIRED BER ON THE RETURN LINK OF THE HYBRID FMT SCENARIO WITH SITE DIVERSITY	195
FIGURE 9.1: CONVERSION FUNCTION FOR KU BAND UPLINK FADING FROM C/N REDUCTION.....	202
FIGURE 9.2: FLOW DIAGRAM OF STATISTICAL ANALYSIS OF FMT AND FIXED MARGIN SCENARIO	203
FIGURE 9.3: FLOW DIAGRAM OF SHORT-TERM SWITCHING ANALYSIS OF FMT SCENARIO.....	204
FIGURE 9.4: ANNUAL AVAILABILITY OF STATIONS	211
FIGURE 9.5: CDF OF ANNUAL AVAILABILITY OF STATIONS.....	212
FIGURE 9.6: PDF OF MF-TDMA RESOURCE REQUIRED FOR ALL STATIONS WITH TRAFFIC TO MAINTAIN QoS PARAMETERS.....	214
FIGURE 9.7: CDF OF MF-TDMA RESOURCE REQUIRED FOR ALL STATIONS WITH TRAFFIC TO MAINTAIN QoS PARAMETERS.....	214
FIGURE 9.8: AN EXAMPLE EVENT FROM THE 8 TH AUGUST 1998 SHOWING A) THE C/N RATIO ON THE FORWARD V BAND LINK, B) THE FMT CONFIGURATION NUMBER OF THE STATIONS DURING THE EVENT AND C) THE PERCENTAGE OF MF-TDMA RESOURCE REQUIRED FOR ALL STATIONS TO OPERATE AT THE SPECIFIED QoS	215
FIGURE 9.9: AN EXAMPLE EVENT FOR STATION 182 FROM THE 8 TH AUGUST 1998 SHOWING A) THE C/N RATIO AND THE C/N RATIO REQUIRED TO MEET QoS REQUIREMENTS, B) THE FMT CONFIGURATION NUMBER AND C) THE BER EVERY SECOND.....	216
FIGURE 9.10: AN EXAMPLE EVENT FROM THE 23 RD DECEMBER 1997 SHOWING A) THE C/N RATIO ON THE FORWARD V BAND LINK, B) THE FMT CONFIGURATION NUMBER OF THE STATIONS DURING THE EVENT AND C) THE PERCENTAGE OF MF-TDMA RESOURCE REQUIRED FOR ALL STATIONS TO OPERATE AT THE SPECIFIED QoS	217
FIGURE 9.11: AN EXAMPLE EVENT FOR STATION 391 FROM THE 23 RD DECEMBER 1997 SHOWING A) THE C/N RATIO AND THE C/N RATIO REQUIRED TO MEET QoS REQUIREMENTS, B) THE FMT CONFIGURATION NUMBER AND C) THE BER EVERY SECOND.....	218

FIGURE 9.12: CDF OF PERCENTAGE OF FMT CONTRIBUTION TO OVERALL MF-TDMA UTILISATION	219
FIGURE 9.13: ANNUAL AVAILABILITY OF STATIONS	223
FIGURE 9.14: CDF OF ANNUAL AVAILABILITY OF STATIONS.....	224
FIGURE 9.15: PDF OF MF-TDMA RESOURCE REQUIRED FOR ALL STATIONS WITH TRAFFIC TO MAINTAIN QoS PARAMETERS.....	225
FIGURE 9.16: CDF OF MF-TDMA RESOURCE REQUIRED FOR ALL STATIONS WITH TRAFFIC TO MAINTAIN QoS PARAMETERS.....	225
FIGURE 9.17: AN EXAMPLE EVENT FROM THE 8 TH AUGUST 1998 SHOWING A) THE C/N RATIO ON THE FORWARD V BAND LINK, B) THE FMT CONFIGURATION NUMBER OF THE STATIONS DURING THE EVENT AND C) THE PERCENTAGE OF MF-TDMA RESOURCE REQUIRED FOR ALL STATIONS TO OPERATE AT THE SPECIFIED QoS	226
FIGURE 9.18: AN EXAMPLE EVENT FOR STATION 182 FROM THE 8 TH AUGUST 1998 SHOWING A) THE C/N RATIO AND THE C/N RATIO REQUIRED TO MEET QoS REQUIREMENTS, B) THE FMT CONFIGURATION NUMBER AND C) THE BER EVERY SECOND.....	227
FIGURE 9.19: AN EXAMPLE EVENT FROM THE 23 RD DECEMBER 1997 SHOWING A) THE C/N RATIO ON THE FORWARD V BAND LINK, B) THE FMT CONFIGURATION NUMBER OF THE STATIONS DURING THE EVENT AND C) THE PERCENTAGE OF MF-TDMA RESOURCE REQUIRED FOR ALL STATIONS TO OPERATE AT THE SPECIFIED QoS	228
FIGURE 9.20: AN EXAMPLE EVENT FOR STATION 391 FROM THE 23 RD DECEMBER 1997 SHOWING A) THE C/N RATIO AND THE C/N RATIO REQUIRED TO MEET QoS REQUIREMENTS, B) THE FMT CONFIGURATION NUMBER AND C) THE BER EVERY SECOND.....	228
FIGURE 9.21: CDF OF PERCENTAGE OF FMT CONTRIBUTION TO OVERALL MF-TDMA UTILISATION	229
FIGURE 9.22: AN EXAMPLE EVENT FOR STATION 534 FROM THE 23 RD DECEMBER 1997 SHOWING A) THE C/N RATIO AND THE REQUIRED C/N RATIO TO MEET QoS REQUIREMENTS ON THE FORWARD TRAFFIC CHANNEL, B) THE BER ON THE FORWARD CONTROL CHANNEL AND C) THE BER ON THE FORWARD TRAFFIC CHANNEL.....	230

Abbreviations and Acronyms

16-PSK	16-ary Phase Shift Keying
ACK UW	Acknowledgement Unique Word
ADALINE	Adaptive Linear Element
APP	<i>A Posteriori</i> Probability
ARMA	Auto-regressive Moving-average
ATM	Asynchronous Transfer Mode
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
BRFM	Bridge Farm Fields
BTP	Burst Time Plan
C/N	Carrier to Noise Ratio
CBTR	Carrier Bit Timing and Recovery
CCLRC	Council for the Central Laboratory of the Research Councils
CDF	Cumulative Distribution Function
CDMA	Code Division Multiple Access
CFM BTP	Capacity and Fade Mitigation Burst Time Plan
C_r	Code Rate
CRC	Cyclic Redundancy Check
DLPC	Downlink Power Control
DSP	Digital Signal Processing
DVB-RCS	Digital Video Broadcasting – Return Channel Satellite
EACK	Entry Acknowledgement
E_b/N_0	Bit-energy per Noise-power Spectral Density
ECS	Entry Contention Slot
EIRP	Effective Isotropic Radiated Power
E_s/N_0	Symbol-energy per Noise-power Spectral Density
FCC	Forward Control Channel
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
FER	Frame Error Rate

FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FLC UW	Forward Link Control Unique Word
FMT	Fade Mitigation Technique
FN	Frame Number
G/T	Gain to Noise Temperature Ratio
GOOD	Goodedge Farm Fields
HDTV	High Definition Television
HYLAS	Highly Adaptable Satellite
HYREX	Hydrological Radar Experiment
IBC	Inbound Carrier
IBS	Inbound Start
IBD	Inbound Duration
ITU-R	International Telecommunications Union – Radiocommunication
LEO	Low Earth Orbit
LLR	Log Likelihood Ratio
LMS	Least Mean Squares
LNA	Low Noise Amplifier
MAC	Media Access Control
MAP	Maximum <i>A Posteriori</i>
MF-TDMA	Multi-frequency Time Division Multiple Access
Mod	Modulation type
NCC	Network Control Centre
NCC FLC	Network Control Centre Forward Link Control
NF	Number of Frames
OBC	Outbound Carrier
OBS	Outbound Start
OBD	Outbound Duration
OSI	Open Systems Interconnection
PDF	Probability Density Function
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quaternary Phase Shift Keying

RAL	Rutherford Appleton Laboratory
$R_{b_{rc}}$	Reference Burst Return Control
RCC	Return Control Channel
RCST RLC	Return Channel Satellite Terminal Return Link Control
RF	Radio Frequency
RMSE	Root Mean Square Error
R_s	Symbol Rate
RSC	Recursive Systematic Convolutional
SFN	Start Frame Number
SNR	Signal to Noise Ratio
SS	Signal to Noise Ratio Sequence
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access
ULPC	Uplink Power Control
VCI	Virtual Channel Identifier
VPI	Virtual Path Identifier
VSAT	Very Small Aperture Terminal

Acknowledgements

Firstly I would like extend my gratitude to my director of studies, Dr Ifiok Otung, for his guidance and support throughout this research. His advice has proven invaluable during the many unforeseen problems that have arisen and without his experience and knowledge of the subject this work would have been considerably more difficult. I would also like to thank my second supervisor Dr Akram Hammoudeh.

I would like to thank all of my colleagues at the University of Glamorgan whose help and support throughout this work have been very much appreciated. I would particularly like to thank Dr Ali Savvaris whose discussions helped me immensely at the beginning of this work and Chethana Nagaraja and Sahena Begum who challenged my understanding of certain concepts when approaching the latter stages.

I would also like to thank the research group at Rutherford Appleton Laboratories and in particular Sarah Callaghan, who was a co-author of one of the publications that resulted from this work and who provided the data and explained the pre-processing procedure.

Finally, I would like to express my deepest gratitude to my parents Paul and Joyce, my sister Kathryn and my grandparents Ellis, Betty, Walford and Joyce for their constant love and support.

Chapter 1

Introduction

1.1 Introduction

Since the launch of the first geostationary telecommunications satellite, INTELSAT I in April 1965, the demand for communication by satellite has grown considerably, encompassing applications such as trunk telephone traffic, for which the first commercial satellites were originally designed, broadcast television and more recently HDTV, large inter-continental corporate networks, mobile telephony and broadband Internet.

One of the more recent satellite applications is broadband internet for which INTELSAT projects a short-term market opportunity exceeding \$10bn in Europe alone [1]. Although satellite broadband provision cannot compete with fibre-optics in terms of bandwidth and has other shortcomings such as increased latency which, without measures such as spoofing or altering the default slow-start mechanisms of the protocol itself [2], can negatively impact upon the efficiency of transport layer protocols, its major advantage is an extensive global coverage that can provide services in areas of low population density that would otherwise not have broadband access.

Satellite broadband has been operating in the *C* and *Ku* bands for some time, by two different methods; two-way schemes provide communication in both directions via satellite while hybrid schemes provide an uplink via a terrestrial link and a downlink via satellite [3].

Although demand for satellite services is growing, utilising bandwidth above the *C* and *Ku* bands, for which a fixed power margin was sufficient to compensate for most

Chapter 1 – Introduction

atmospheric fading, became more complex, requiring fade mitigation techniques (FMTs) to compensate for rain induced fading. Satellite broadband provision in the *Ka* band has recently become a reality with services such as “tooway” [4], which uses adaptive signal processing to provide broadband Internet via Eutelsat satellites to home users. The bandwidth available at *Ka* band may satisfy current demand but when this bandwidth approaches saturation the *V* band may eventually become an attractive resource, provided it can be successfully exploited. Unfortunately, the atmospheric obstacles that render *Ka* band communication more complex affect the *V* band even more drastically. This applies to every source of atmospheric degradation that is present at *Ku* band, most notably for rain induced fading [5], but also for gaseous absorption [6] and scintillation [7], with a significant addition by cloud [8] that can be neglected at *Ka* band frequencies and below. Figure 1.1 shows the fixed power margin that would be required for increasing frequency to achieve availabilities of 99%, 99.9%, 99.99% and 99.999% to compensate for rain alone. These results are obtained from the ITU-R model for annual attenuation due to rain [9].

Although heavily outweighed by the myriad of disadvantages, *V* band communication does benefit from a higher antenna gain and narrower beamwidth which can be exploited in a frequency reuse scheme with several narrow spot beams.

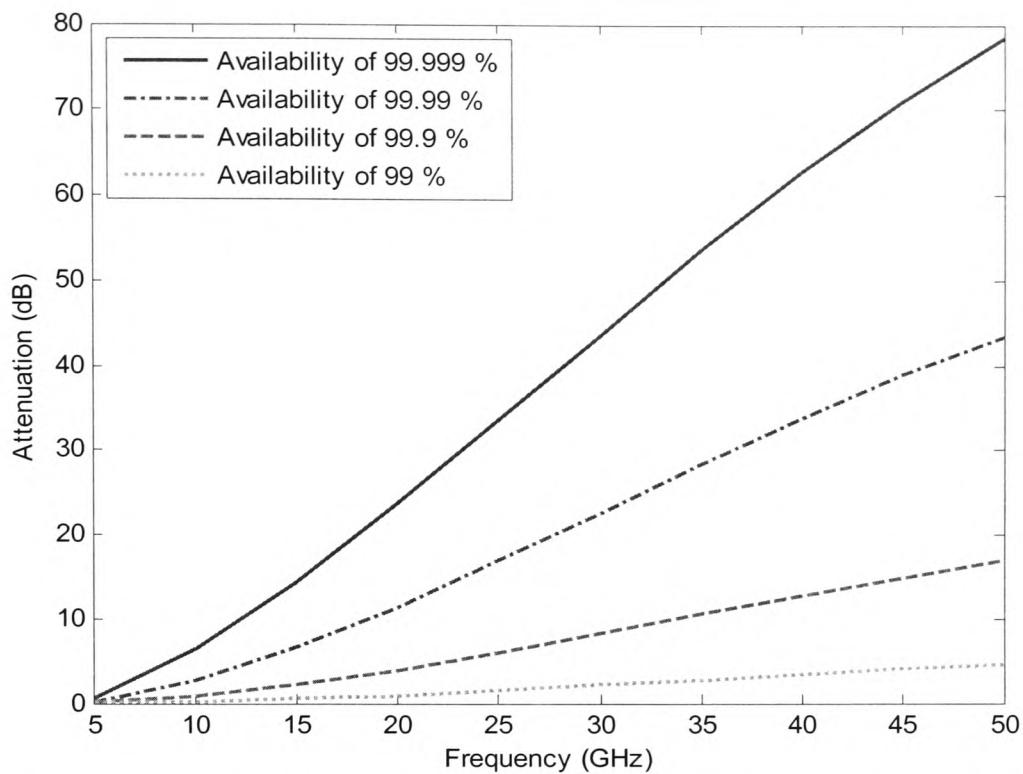


Figure 1.1: Rain attenuation not exceeded for a specified availability as a function of frequency

The principal aim of this work was to investigate the various aspects of V band satellite communication and develop a strategy towards an enabling technology to be employed at these frequencies. The particular frequencies under consideration in this thesis are shown in table 1.1.

Frequency Band (GHz)	Band Identifier	Frequency Range of Interest (GHz)	Bandwidth Allocated for Fixed Satellite Services (GHz) [10]
3.9 – 8.5	<i>C</i>	N/a	2.545
10.9 – 17.2	<i>Ku</i>	11.35 – 13.87	3.350
36 – 46	<i>Q</i>	39.52 – 39.56	3
46 – 56	<i>V</i>	50.42 – 50.46	4

Table 1.1: Frequencies and associated bands under consideration

Chapter 1 – Introduction

Of the many types of communication available, broadband Internet provision to home users was chosen as the case scenario due to the widespread use of the Internet and the relatively high demand for this particular service. Figure 1.2 shows the system architecture under consideration.

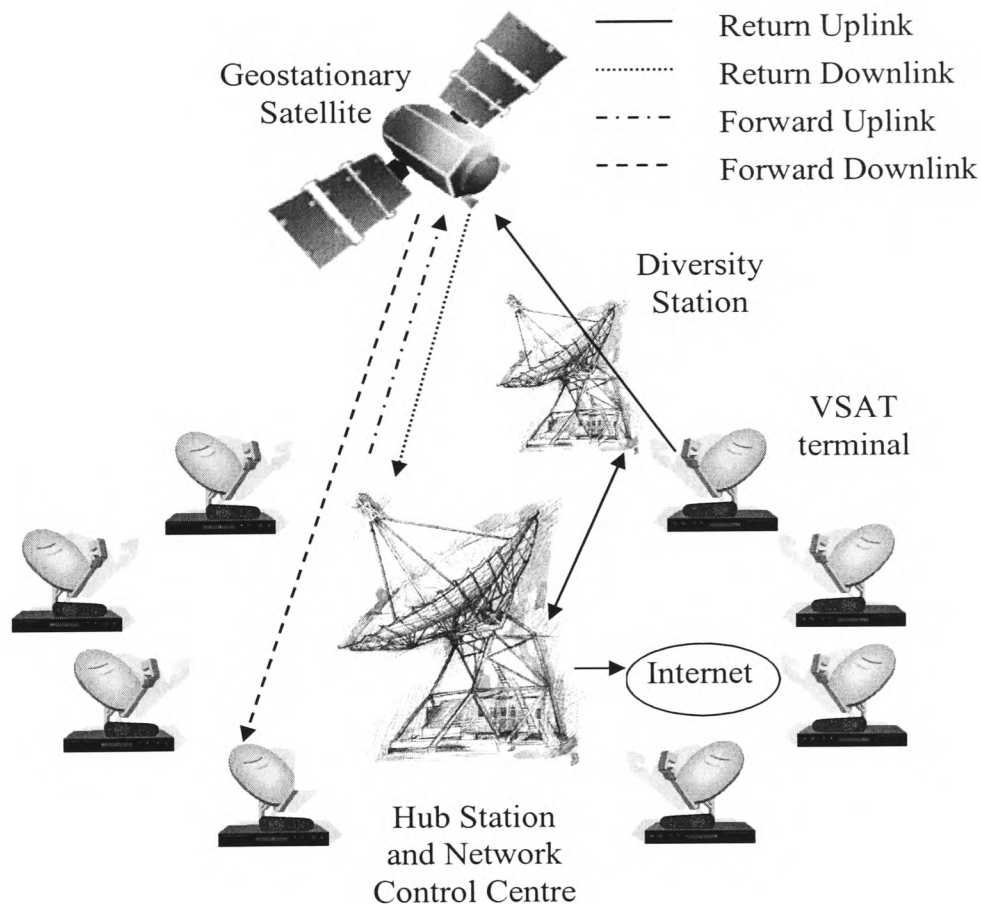


Figure 1.2: System Architecture

1.2 Objectives of Research

In order for *V* band to be successfully exploited should the demand rise and necessitate its eventual use, the precise effects of atmospheric fading must be understood, not only in terms of annual statistics but alongside many of the other factors involved in

Chapter 1 – Introduction

implementing an FMT controller. While the broader objective was to develop a system that could provide broadband services to home users in rural or otherwise sparsely populated areas with an acceptable availability, several distinct stages with their own particular objectives can be identified as component parts of the larger picture.

Firstly, in order to fully characterise the problem, a detailed review of the first and second order fade dynamics present at V band was undertaken. The objective of this first stage was to understand the specifics of the problem in question, before attempting to solve it, an engineering truism.

Secondly, various methods of short-term fade prediction were investigated with the intention of being included as a component part of a fade controller. The performance of any controller will be affected by the accuracy of each of its component parts, a short-term predictor being one of these. The objective here was to minimize the inaccuracy of this component. This led to the development of a new hybrid predictor.

A third objective was to process a period of data collected by Rutherford Appleton Laboratory (RAL) in order for it to be used in a simulation. This involved removing days from the sequence that contained a large amount of invalid data and either frequency scaling or interpolating smaller sections of invalid data. The objective here was to create a reliable time series of attenuation to be used in the final simulation, avoiding undesirable contributions from invalid data which would give spurious results in the final stages. This yielded a time series of 193 days for the final stage.

Chapter 1 – Introduction

In order to simulate the various FMTs, they each had to be coded in Matlab. This constituted the fourth objective, and while no innovation resulted from the implementation of the DVB-RCS turbo code [11], a novel procedure was developed for site diversity estimation from a single time series of attenuation.

A fifth objective was to design a protocol to provide a higher availability to control information than traffic information.

Lastly, the protocol mentioned previously was subsequently used to design link budgets and these were in turn used in a simulation of a satellite broadband Internet scenario, the objective being to show whether or not V band communication is possible using a variety of FMTs.

The field of satellite Internet is fairly broad and it is perhaps worth mentioning what were not the objectives of this work. Firstly, the vast majority of the work is concerned with the physical and data-link layers of the OSI model [12]. While some attempt has been made to include traffic, it is not the primary concern. Likewise, the protocol design is not intended to be a generic protocol with all the complexities of that found in [11], neither has an attempt been made to compare it to other protocols designed to work in lower frequency bands such as [13]. It is merely a means to create a meaningful link budget and to protect control information from the insecurities of communication at V band. Also, there are many areas of current research into methods of improving communication such as satellites with on-board processing, adaptive beam-shaping or on-board switching. To take a holistic approach would be extremely time consuming

Chapter 1 – Introduction

and therefore a simple comparison with the fixed link margin method of satellite link design is used throughout.

1.3 Overview – Content of Thesis

Following this introductory chapter, the remainder of the thesis is laid out in chapters that describe the work undertaken in the most suitable order, which is almost chronological but for a few exceptions.

Chapter 2 gives a literature review of FMTs, highlighting the various ways in which signal degradation due to rain and other atmospheric effects can be overcome. The advantages and disadvantages of each with respect to a satellite broadband scenario are discussed with a final decision on the ones that are utilized in the later stages of the work. This chapter also gives a literature review of fade control, with a generic fade controller outlined which serves as a basis for the work in the subsequent chapters.

Chapter 3 discusses fade dynamics at V band, including a review of work on fade dynamics by other authors as well as a detailed discussion of the results of second order fade dynamics analysis carried out on the attenuation time series provided by RAL. The analyses include fade slope, fade duration and inter-fade duration.

Chapter 4 gives a detailed description of the turbo code that was used as a forward error correction (FEC) scheme, both for the coding and decoding procedure and its performance for code rates and block sizes relevant to the later chapters.

Chapter 1 – Introduction

Chapter 5 documents the procedure and results of a site diversity approximation, using a single time series. An optimal pair of diversity stations was chosen from the rain gauge measurement campaign of the Hydrological RADAR Experiment (HYREX) [14] with respect to the ITALSAT satellite from which the time series of attenuation was gathered. The proportion of heavy rain and light rain at both sites was measured and using the rain rate thresholds for these two categories of rain, the corresponding link attenuation was calculated with the ITU-R model for specific attenuation [15]. Following this, results were obtained for the proportionality of heavy rain and light rain at separate points in time, for different temporal separation distances. The time separation that gave results that fitted most closely with the rain gauge results was chosen as a near-equivalent to separation distance and results obtained and compared with the ITU-R model for site diversity [9].

Chapter 6 gives a background to short-term prediction of attenuation and proposes an adaptive linear element (ADALINE) method for this prediction. A hybrid predictor of both an ADALINE and an autoregressive moving average (ARMA) method is compared with other types of predictor from the literature and the results presented.

Chapter 7 outlines the preliminary design that had to be undertaken before the final simulations could take place. This includes the dual-channel protocol that was required to create meaningful link budgets and protect control information, traffic considerations and the data pre-processing that was carried out in order to obtain a reliable set of data for simulation.

Chapter 1 – Introduction

Chapter 8 gives the link budgets, methodology and results of a preliminary investigation of performance of combined FMT scenarios. This was done with several simplifying assumptions to reduce coding and simulation time and to take any information obtained into the final simulation stage.

Chapter 9 gives the link budgets, methodology and results of the final simulation scenario. This includes both a long-term statistical analysis and a short-term switching analysis of various events. Also, two scenarios are investigated with different levels of each FMT to estimate how much resource should be devoted to each.

Finally, chapter 10 concludes the thesis, summarising the work as a whole and giving recommendations for further work. Figure 1.3 shows the relationship between the chapters, grouping some into broader topics to show the overall aim of the work.

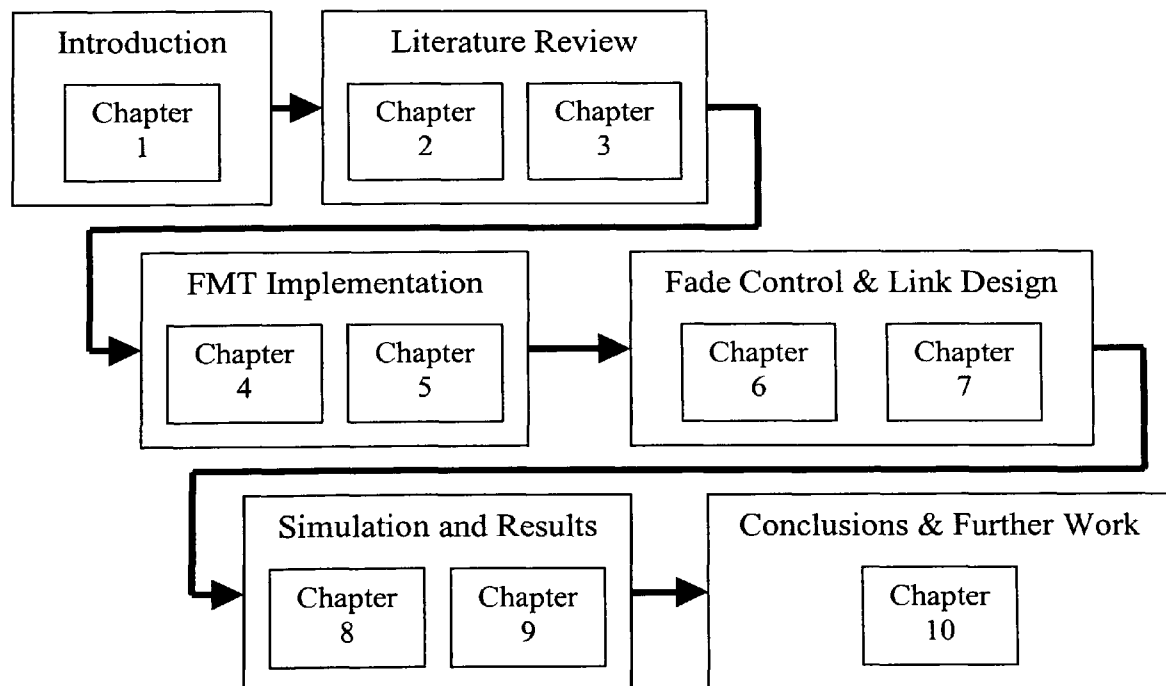


Figure 1.3: Diagram of thesis layout

1.4 Summary of Novel Contribution

The novel contribution of this thesis is summarised below:

- A detailed analysis of second order fade dynamics at V band and a comparison of fade slope at V band to the current ITU-R model [16]. This work was published in [17] and [18].
- A method of reproducing site diversity with a single time series of attenuation.
- An ADALINE short-term fade predictor. Although the ADALINE method is not new, the application is. This work was published in [19] and [20].
- A hybrid short-term fade predictor. This work was published in [21].
- A dual-channel protocol suitable for V band communication.
- Results of a simulated satellite broadband Internet scenario that utilizes several FMTs as an enabling technology for V band communication.

Chapter 2

Fade Mitigation and Control

2.1 Introduction

Before the recent commercial utilization of the *Ka* band, satellite systems were designed with fixed power margins to ensure availability for a specified percentage of time. The reasons that this method was not suitable at *Ka* band were twofold, firstly due to the amount of wastage that would result from communicating at all times with a power margin in the region of 10 to 20 dB (see figure 1.1) and secondly due to regulatory restrictions on the power flux density on the Earth that are in place to avoid interference with other satellites.

At *Ka* band, the issue of FMTs underwent much research, leading to a vast field of scientific journal papers and several theses [22, 23]. Recently this has led to the deployment of satellites with *Ka* band transponders for commercial use such as Eutelsat's HOT BIRD and ATLANTIC BIRD [24] as well as proposals for advanced satellites that operate exclusively in the *Ka* band such as HYLAS [25].

At *V* band, not only are the power flux density regulations still in place and the problem of wasted resources compounded, but the level of extra power required for communication at an availability approaching that of services in the lower bands is simply too high to be realised in practice (see chapter 7). Large increases in antenna size are not a practical option for affordable systems and would be contrary to the trend of decreasing antenna sizes, not to mention being the very opposite of the definition of a VSAT.

Chapter 2 – Fade Mitigation and Control

To prevent a theoretical V band system being limited to regions where there is very little rain, or to commercially unviable earth stations, an FMT scheme is required that can overcome larger attenuations due to rain. From figure 1.1, for the same outage probability, the dynamic range requirement for V band is approximately 20 dB above what is necessary even at Ka band.

In order to successfully implement FMTs, a fade controller is also required. This must be able to efficiently switch between different FMT settings while maintaining a certain maximum allowable bit error rate (BER), which is one principal component in the quality of service (QoS) parameters of any communications link.

The remainder of this chapter reviews each of the various types of FMT, highlighting the tradeoffs involved and detailing the advantages and disadvantages with respect to the VSAT broadband Internet scenario that is of particular interest in this thesis. A final selection of those FMTs that are suitable for a satellite broadband Internet scenario is made. Fade mitigation control is also reviewed briefly, and a generic fade controller is reviewed which serves as a basis for the remaining work throughout this thesis.

2.2 *Fade Mitigation*

2.2.1 *Introduction*

The field of fade mitigation has featured extensively in the recent literature and can be roughly divided into four main categories. Firstly there are signal processing techniques such as adaptive FEC coding, adaptive modulation schemes and data rate reduction, whose aim is to alter the mode in which the data is sent at the physical layer. Secondly

Chapter 2 – Fade Mitigation and Control

there is diversity which aims to avoid an attenuated link by a re-routing strategy. Examples of this are site diversity or orbital diversity where the aim is to re-route the signal in the spatial domain to physically avoid a rainstorm or frequency diversity which attempts to re-route the communication in the frequency domain in order to reduce fading to a manageable level. Thirdly there are adaptive multiple access schemes such as adaptive TDMA or adaptive CDMA which aim to allocate a certain common resource to users with fading conditions. This is particularly useful in combination with the signal processing techniques already mentioned. Lastly there is power control which simply aims to increase the transmit power of the link in proportion with the C/N reduction. Each of these techniques will be reviewed in the following sections.

2.2.2 Signal Processing

2.2.2.1 Adaptive Modulation

Adaptive modulation aims to maintain a suitable bit error rate (BER) by altering the modulation scheme depending on the channel conditions. Nominally the system would operate in the maximum spectral efficiency configuration for clear sky conditions and would switch to schemes with lower throughput that require less bit energy per noise power spectral density (E_b/N_0) to maintain the equivalent BER during fades. The uncoded BER performance of typical modulation schemes used in this method are illustrated in figure 2.1.

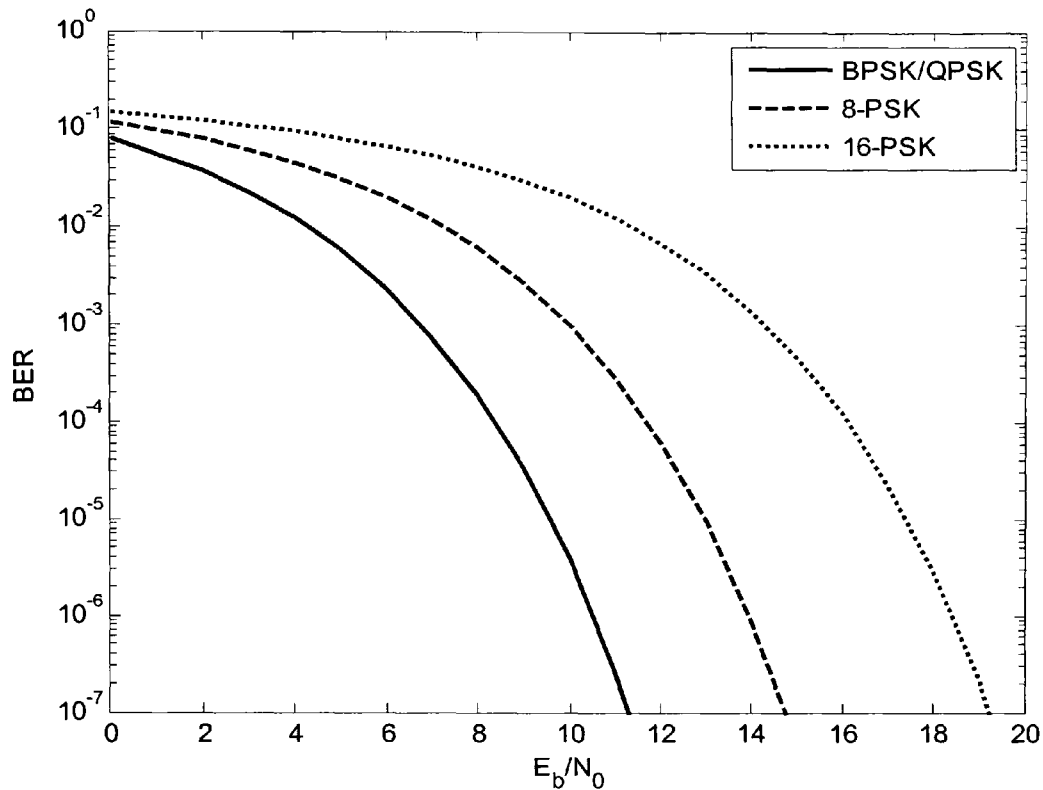


Figure 2.1: BER performance of uncoded PSK schemes

An early study on adaptive modulation in [26] compared various phase shift keying (PSK) and quadrature amplitude modulation (QAM) schemes by means of a prototype modulator that was successfully built and tested. The authors conclude that while a demodulator is more complex, it is still feasible by use of a universal carrier tracking loop. Following this there have been other studies which make use of adaptive modulation, such as [27] in which a DSP simulator is used to switch between modulation schemes from BPSK to 16-PSK, [28] in which a Markov chain is used to switch between modulation schemes ranging from BPSK to 16-PSK, and [29] in which a neural network is used to switch between modulation schemes ranging from BPSK to 16-PSK.

Chapter 2 – Fade Mitigation and Control

However, there are two major drawbacks involved with high-order modulation schemes. The first is problematic to satellite communications in general and the second is application specific to a broadband Internet return link.

The first difficulty arises from the nonlinear nature of the so-called bent-pipe satellite transponder, which can cause a far greater amount of incorrectly received symbols than the purely theoretical BER curves suggest [30, 31].

The second difficulty arises from the bi-directional nature of the link and the vastly different earth station power amplifier outputs for each direction. As a result, although high-order modulations are possible in the forward direction, with a large and expensive power amplifier at the hub, in the return direction there is a severe power limitation, which prohibits their use. Since attenuation has to be overcome both in the forward and return directions, higher order modulations are not possible without making the FMT schemes on both directions asymmetrical, which would not only complicate matters but leave both directions with different dynamic ranges which would ultimately limit availability to the capabilities of the weaker link regardless.

This second problem became clear while designing the link budgets (see chapter 7). However, it was still possible to make use of a more limited form of adaptive modulation that utilised QPSK in clear-air conditions and BPSK once a certain fading condition had been met.

2.2.2.2 Adaptive Coding

Adaptive FEC coding aims to maintain a suitable BER by adding redundancy to the digital information as channel conditions degrade. This can be achieved with a bank of different coders and decoders, each designed according to certain channel conditions or more simply by puncturing convolutional or turbo codes so that certain bits in the generated code are omitted at the transmitter and replaced at the receiver. Fixed codes have already seen widespread use in satellite communications, enabling some coding gain with respect to the bit-energy per noise power spectral density (E_b/N_0) required to achieve a specified BER, as shown in figure 2.2. This provides a useful trade-off between C/N requirement and bandwidth. The aim of adaptive codes is to operate with a low level of redundancy in clear-sky conditions with redundancy increasing in stages as fading increases in order to maintain the BER at an acceptable level.

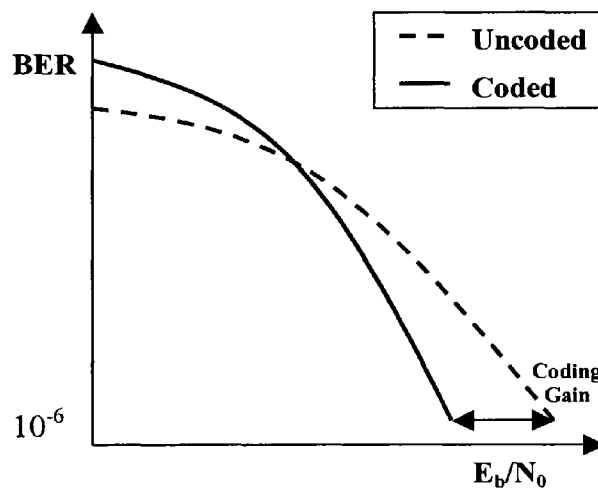


Figure 2.2: FEC coding gain

Serially concatenated FEC codes are commonly used in satellite communications, a typical example being a Reed-Solomon outer code with a convolutional inner code and an interleaving stage between the two [10]. However, since the inception of turbo codes in the early 1990s [32], a great deal of coding gain can be achieved in comparison with

the traditional method [33], although it is possible to serially concatenate turbo codes with Reed-Solomon codes for an even greater performance [34]. Turbo coding has already been standardized for DVB-RCS [11], which uses a duo-binary coding technique to encode two bits at a time with a variable puncture rate that allows several code rates with several block sizes also supported, most notably a 53 byte block for direct use with ATM cells. Other work is also in progress for optimisation of turbo coding for the satellite transmission channel [34] and higher order modulations [35].

Provided the transmission medium can support variable resource allocation to account for changes in bit-rate, the technique is particularly suitable to satellite broadband Internet. As with most signal processing methods it is relatively cheap to implement in comparison to other methods, its only drawback being a decoding delay caused by the fact that a block of information of predefined length must be received before decoding can take place and that several decoding iterations must be completed to achieve the improved gains mentioned previously. However, the increased latency is small in terms of packet arrival times for terrestrial networks and negligible when compared with the round trip delay of a geostationary satellite. For this reason adaptive coding was selected as one of the methods for implementation in a satellite broadband Internet scenario. More specifically, the DVB-RCS turbo-coding scheme was selected, as duo-binary codes require fewer decoding iterations than binary codes, leading to smaller delays. Several code rates are already available, ranging from $1/3$ to $6/7$, as are block sizes, one of which supports the ATM cell format, with appropriate interleaving algorithms in each case.

However, while this coding scheme is very attractive, it allows for a dynamic range of merely 3.5 dB, far short of the requirement for maintaining an acceptable availability at V band and cannot therefore be considered as an acceptable solitary solution to the problem.

2.2.2.3 Data Rate Reduction

The term data rate reduction has been used to signify different methods by different authors throughout the literature. The various methods will be described and given more specific titles in order to clarify the subject as far as possible.

The first definition is referred to here as direct bit-rate reduction, such as that proposed in [36], whereby the information bit-rate is reduced directly regardless of the resource sharing mechanism, and without any other action being taken. This leads to a reduced operating bandwidth, which in turn leads to a reduction in noise power, thereby reducing the E_b/N_0 . This method was briefly investigated during the final simulation stages but was found unsuitable as although the noise power is reduced on both the uplink and downlink, the downlink transmitted power is also decreased since the occupied bandwidth is narrower, degrading the downlink section and unbalancing the overall link. This problem could possibly be avoided by use of a satellite with on-board processing that could alter power allocation to different links, but within the scope of this work where a bent-pipe transponder is investigated exclusively, the method cannot be utilised.

A second definition found in [23] refers to spread spectrum techniques as data rate reduction. This is also known as adaptive fade spreading [22], and will be referred to as

such in this work. The technique involves reducing the information bit-rate during fades while maintaining a constant pseudo-random chip-rate, which spreads the information by a factor proportional to the chip-rate to bit-rate ratio while maintaining a constant bandwidth unlike the previous method. A study in [37] uses this technique for videoconferencing purposes with video codecs of 2.048 Mbps down to 256 kbps for different fading levels and notes that it is only suitable for services in which data rate reductions can be tolerated. While this is not a theoretical problem for Internet services it has not been utilised in this work due to a second more restrictive problem where, due to the spread-spectrum nature of the technique, the E_b/N_0 at the demodulator is small, having a negative impact on phase recovery [38]. This problem is compounded when considered alongside a resource sharing mechanism such as TDMA or MF-TDMA, which requires higher transmission rates, particularly for the return channel which is power limited.

A third and fourth definition that apply exclusively to adaptive resource sharing techniques are also found in the literature. These will be referred to as adaptive resource reduction and adaptive burst duration and will be described in the following section.

2.2.3 Adaptive Resource Sharing

2.2.3.1 Adaptive TDMA

Adaptive TDMA involves the use of a common resource in any given TDMA frame being allocated to users with fading conditions [39]. This method can be used alone or in combination with data rate reduction, adaptive modulation or adaptive coding.

Chapter 2 – Fade Mitigation and Control

Adaptive resource reduction is a method where the information bit-rate of the user is reduced, in turn reducing the amount of space required in the TDMA frame. This is not strictly an FMT, as no change in the link results from this action, although it could be a useful tool for resource management, especially for hub fades where all users are affected equally. However, since for this work we are interested in FMT performance, this technique has not been implemented.

Adaptive burst duration is a method in which the information bit rate remains constant but a longer time slot is allocated to the station. This extra time can be utilised in one of two ways, either by data rate reduction where the burst rate within the TDMA frame is reduced, or by increasing redundancy via a higher coding rate, both methods maintaining a constant information bit-rate. However the former suffers from the same problem as the direct bit-rate reduction discussed in the previous chapter, whereby the operating bandwidth decreases, in turn decreasing the noise power on both uplink and downlink but also decreasing the transmitted power on the downlink. A far better alternative is the combination with adaptive coding, which allows a constant bandwidth to be maintained, while the BER criterion for acceptable QoS is still met at a lower E_b/N_0 .

The satellite broadband Internet scenario in question is not particularly well suited to an adaptive TDMA scheme since the bi-directional link is power limited in the return direction. However, a similar scheme using MF-TDMA can also be implemented, such as that in [40], reducing the burst rate significantly in the return direction. A combined TDMA and MF-TDMA scenario was eventually chosen for the forward and return links respectively. While some researchers use the idea of fixed reserve slots, the idea of

customisable burst duration, such as that supported in the DVB-RCS standard [11], is used in this work. This allows the frame to be defined precisely; each station is given a precise burst start time with reference to the start of the frame and precise burst duration. This supports an adaptive coding scenario more efficiently than the idea of fixed size slots, which lend themselves better to a data rate reduction scheme, such as that in [41], the problems of which have already been discussed.

2.2.3.2 Adaptive CDMA

Adaptive CDMA has undergone far less research than adaptive TDMA, but some studies have been carried out such as that in [42]. The technique reserves a common resource that may be allocated to stations with fading conditions. Stations without fading are assigned a single code from a quasi-orthogonal set, with several of these spreading codes reserved until any station experiences fading. The extra codes can then be assigned to the station and a variable number of bits can be mapped onto the multiple available sequences.

While the authors report good results in comparison with fixed CDMA access schemes, CDMA has not been implemented as an access scheme due to the higher efficiency of TDMA. CDMA is more appropriate in a multipath environment, such as a low earth orbit (LEO) satellite constellation for mobile communications.

2.2.4 Diversity

2.2.4.1 Site Diversity

Site diversity attempts to mitigate fading by the use of two or more earth stations that share a terrestrial link to one another. The rationale behind site diversity is that the two

stations are sufficiently separated so that the likelihood of a convective rainstorm being situated concurrently in the earth-space path of both stations is small [43] as illustrated in figure 2.3. Attenuation measurements at both sites are made and switching takes place when the active station fading level exceeds that of the inactive station. Some hysteresis may also be necessary to avoid repeated switching if the changeover carries some overhead [44, 45].



Figure 2.3: Site diversity illustration

Site diversity can be characterised either by the amount of diversity gain or diversity improvement that they provide. Diversity gain is the reduction in EIRP that would be required for an equivalent availability if a diversity station were used. Diversity improvement is the increase in availability that would occur if the EIRP were to remain constant and a diversity station used. These concepts are introduced in figure 2.4, which shows the CDF of availability for a single station in southern England and the equivalent CDF with a site diversity station added at a distance of 10 km and a baseline orientation of 90° from the propagation path. These are both generated from the ITU-R model [9]. The CDF in this figure and other figures throughout this work uses the

probability of exceedance, beginning at 100 percent. This is a standard way of presenting results for the scenarios under consideration as used by the ITU-R.

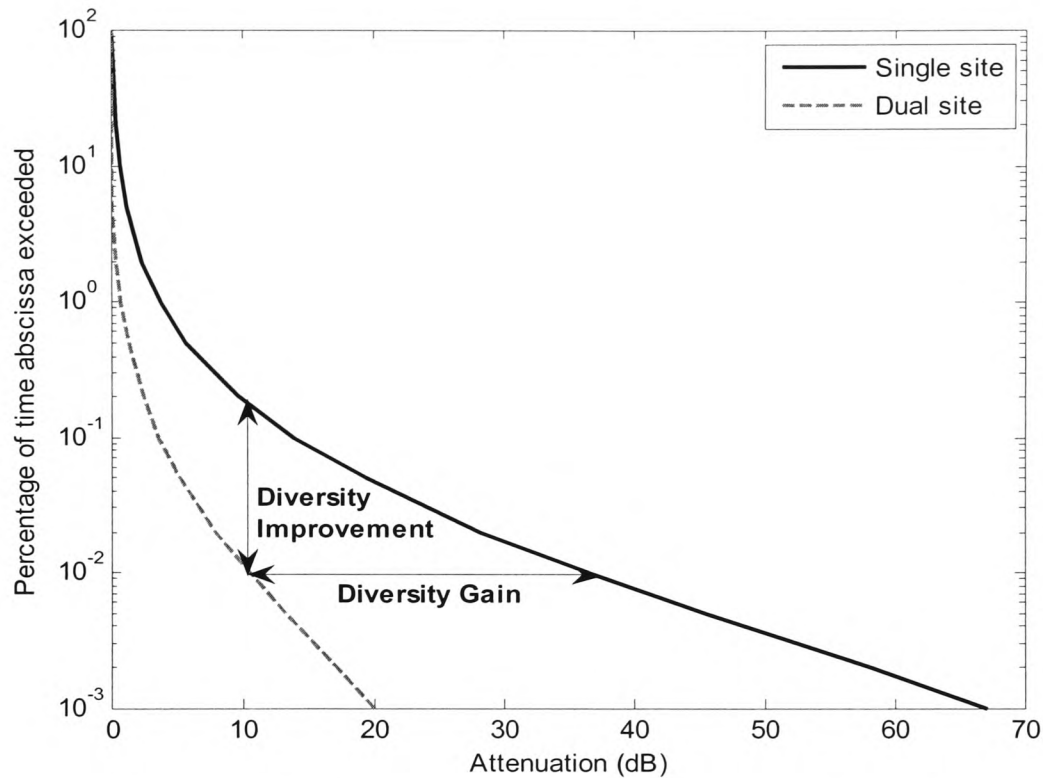


Figure 2.4: CDF of annual rain attenuation exceeded for a single and dual site scenario.

A major advantage of this technique is that no change is required by the user terminal at all, the physical layer mechanisms remain the same and there is no change in bit-rates, bandwidth or burst duration. However, the fixed wired link that is necessary for this and the cost of a second earth station can be prohibitive for VSATs. A study has been carried out on the suitability of short-scale diversity with application to VSATs [46], with encouraging results for a network of VSATs in close proximity, although this scenario directly conflicts with the main benefit of broadband Internet via satellite, namely the near-ubiquitous coverage and possibility to provide services in remote areas with low population density.

Site diversity is in fact a perfect complement to the signal processing techniques discussed previously. While the signal processing techniques are able to mitigate the effects of fades at individual VSATs, these methods are incapable of providing protection for every link that would be affected in the event of hub fading due to the limited common resource. Site diversity is an excellent way of avoiding the drastic fading that occurs as a result of convective rainstorms at the hub without affecting the resource sharing properties of the system. Therefore it has been implemented for the hub side only, a description of which is given in chapter 5.

2.2.4.2 Orbital Diversity

Orbital diversity also attempts to re-route the signal, but unlike site diversity uses 2 or more satellites, as illustrated in figure 2.5, the diversity gain improving as the angular separation between them increases. Instead of making use of the statistical size limit of convective rainstorms, this method makes use of the decorrelation of the path of the rainstorm in relation to two or more satellites, i.e. as the cell moves into or out of the path of one satellite, the other is used. For this reason, orbital diversity does not give gains as high as site diversity since the cell is less likely to be totally avoided [47].

Orbital diversity has similar strengths and weaknesses as site diversity, the advantages being unchanged information bit rate and bandwidth and no on-board processing requirement. The major disadvantage is the increased cost incurred by the use of extra satellites and antennas, although this is not quite as high as site diversity since no terrestrial link is required. Several measurements and models have been made on orbital

diversity [48, 49], although the subject has been less thoroughly investigated than site diversity.



Figure 2.5: Orbital diversity illustration

Since site diversity was chosen as a principal component of hub fades for the broadband Internet scenario, orbital diversity has been avoided. This is primarily due to its lower gain in comparison with site diversity [47] rather than any implementation problem preventing its use.

2.2.4.3 Frequency Diversity

Unlike site diversity and orbital diversity, frequency diversity involves re-routing the signal by a frequency difference rather than a spatial difference [50]. Lower frequencies are less attenuated by rain and can be utilised by terminals that are experiencing rain-induced fading.

An analysis of a frequency diversity system with *Ka*-band handling the bulk of the traffic and various amount of shared bandwidth at *Ku*-band being allocated has been carried out [51]. For an availability of 99.98% and a reserve bandwidth of 10% at *Ku*-band, the rain power margin required for *Ka*-band communication dropped from over 24 dB to 8 dB in this system in addition to the earth station antenna diameter dropping from 11 metres to 4 metres. Given the scaling of attenuation, this improvement factor increases when a *V* band system is implemented with a *Ku* band shared resource. This is shown in figure 2.6 where, using the ITU-R model for rain attenuation [9], a 12 GHz CDF of rain attenuation is compared with that of a 40 GHz CDF in a downlink frequency diversity scenario and a 14 GHz CDF of rain attenuation is compared with that of a 50 GHz CDF in an uplink frequency diversity scenario. The graphs from the model indicate a diversity gain of roughly 24 dB at the downlink frequencies and 31 dB at the uplink frequencies, assuming that a sufficient common resource is provided to accommodate all fading stations.

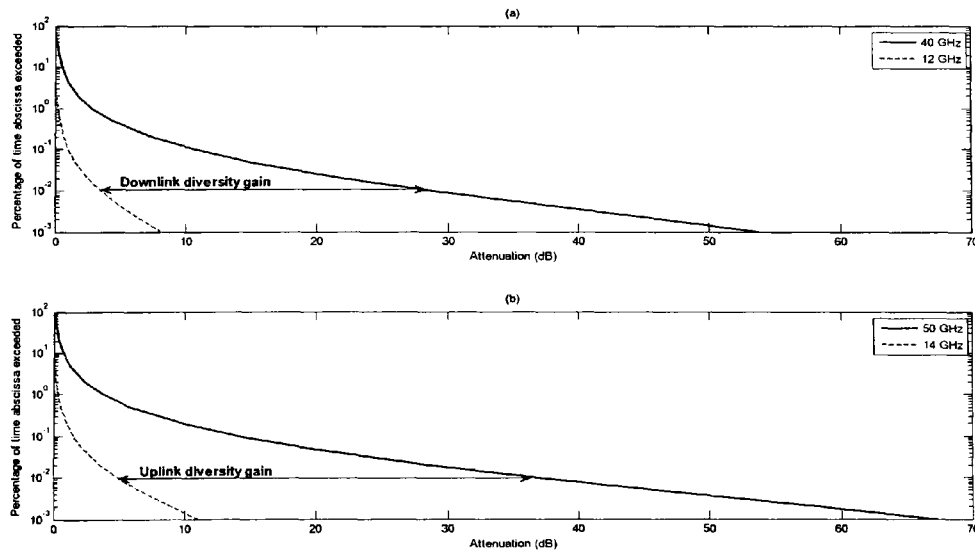


Figure 2.6: CDFs of frequency diversity showing a) downlink diversity gain and b) uplink diversity gain for a V band communications system with a *Ku* band shared resource

Frequency diversity has the same advantages as other forms of diversity, i.e. no on-board processing is required and no change in user bit rate or bandwidth, but it also has the additional advantage that fades are reduced on both the uplink and downlink in both directions. Another potential advantage is that signalling information can be sent via the lower frequency band, guaranteeing a higher signalling quality than data quality. This is explored further in chapter 7.

However, frequency diversity has the disadvantage of increased terminal cost as two different sets of RF equipment are required at every earth station, which is a similar constraint as site diversity, but since the RF equipment is installed at the same site, no physical link is required as with site diversity. A further disadvantage is that, assuming a single antenna is used at both frequencies, link budget design becomes a more complicated process since antenna gains are not equal at both frequencies and this can lead to a process of much iteration with asymmetrical transmission powers and different code rates for each link.

2.2.5 Power Control

2.2.5.1 Uplink Power Control

Uplink Power Control (ULPC) aims to overcome fading on the uplink by monitoring the fading conditions, usually by measuring an unmodulated carrier close to the downlink frequency and using frequency scaling to estimate the uplink attenuation. Other methods exist but the beacon measurement with downlink monitoring is most cost-effective as a single low noise amplifier (LNA) can be used for both modulated carrier and beacon measurement. However, it may not always be possible to measure a

beacon frequency close to the downlink frequency with commercial satellites and other methods may be required, such as measuring the sample mean and variance of symbols from a known sequence at the output of a correlation receiver [52]. It has been claimed that this closed-loop method is unsuitable for ULPC [53], but if the known sequence originates at the hub on the forward link, the overall C/N reduction at the hub for both the uplink and downlink can be measured and a look-up table used to determine the contribution of uplink fading. This method has a longer time interval and is therefore less efficient than direct monitoring of a beacon signal but is practicable nonetheless. Another method of closed-loop power control is also possible by resending fading values back and forth between stations, as investigated in [54], although this has been avoided here as many VSAT links could incur a considerable overhead.

ULPC has been well researched, with many experiments and simulations having been carried out, such as power control as an exclusive FMT [55], but also as a joint method of compensation with other FMTs such as data rate reduction [37] or diversity [56, 57].

For the broadband Internet scenario considered here, ULPC is a particularly useful technique which complements site diversity well and has been implemented at the hub in the final stages of this work.

2.2.5.2 Downlink Power Control

Downlink power control (DLPC) aims to reduce the effects of downlink fading by allocating some reserve power on board the satellite itself [58]. However, the power available on board satellites is limited, and any margin would have to be small in comparison to uplink power control, where power itself is less of an issue. In addition,

DLPC would also require some on-board processing to function and would be difficult to design in a frequency-sharing scheme such as FDMA or MF-TDMA as the inter-modulation products would change according to the amount of back-off applied to the LNA. For these reasons it is a less desirable option than ULPC and has not been included in this work since onboard processing has not been considered.

2.2.5.3 On-board Beam Shaping

On-board beam shaping attempts to increase EIRP to areas within the satellite footprint that are experiencing rain [59]. This requires multiple spot beam antennas that are able to adjust the shape of the beam according to information about rain activity. In order for this to be implemented, on-board processing would be necessary, in addition to some control centre processing of satellite imagery to estimate the rain activity in different regions [60], [61].

While this is a possibility for broadcasting, it is of little use to VSAT networks that operate on a point-to-point basis and has not been considered for the case scenario in this work.

2.2.6 FMT Scenario Overview

The FMTs utilised in the later stages of this work are summarised in table 2.1 below. Each method is characterised in terms of the terminal for which it provides compensation, the amount of compensation provided and its comparative cost [62]. A system architecture diagram of the schemes is shown in figure 2.7.

Name	Intended Terminal	Estimate of Gain (dB)	Comparative Cost
Adaptive Modulation	VSAT	3	Low
Adaptive Coding	VSAT	3.5	Low
Adaptive MF-TDMA	VSAT	N/A (used with modulation and coding only, no data rate reduction)	Low
Site Diversity	Hub	26	High
Frequency Diversity	VSAT	24	High
ULPC	Hub	6	Low

Table 2.1: Summary of utilised FMT techniques

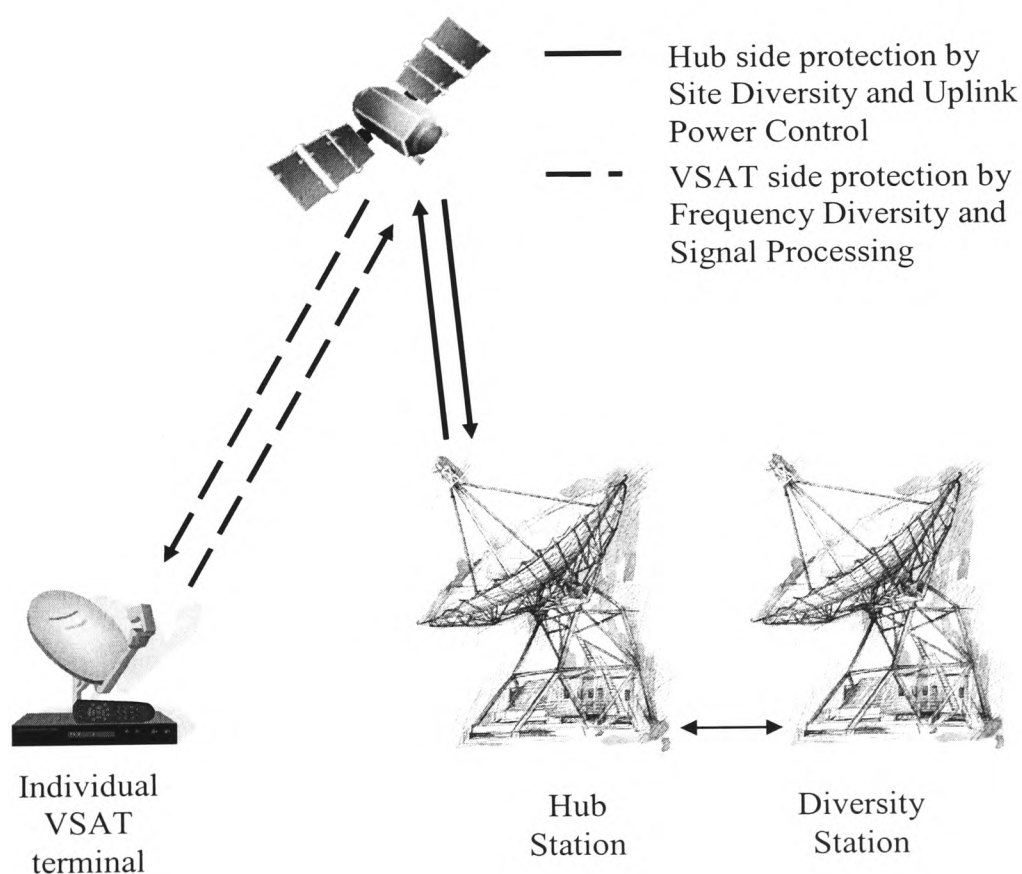


Figure 2.7: FMTs with respect to system architecture

2.3 Fade Control

2.3.1 Introduction

The object of fade mitigation control is to provide the resources necessary to enable acceptable QoS during fade events, while concurrently attempting to optimize the use of system resources by avoiding overcompensation and unnecessary switchovers. This can be divided into four main categories:

- Fade detection
- Short-term fade prediction
- Resource allocation
- Protocol design

These four categories are discussed in subsequent sections, following a discussion of a generic fade controller.

2.3.2 A Generic Fade Controller

A generic fade controller was presented in [53] and is reproduced here in figure 2.8. The fade detection, short-term prediction and later decision and activation blocks are discussed under the four categories in later sections. However the requirement for duration, inter-duration and peak inputs must be discussed for a complete analysis, along with fade slope.

The combination of these inputs depends to a large extent on the protocol used. If no knowledge of the current C/N ratio is available, by beacon measurement or otherwise,

then statistical components may play a significant part in a fade control system since no real-time information can be used. However, in the presence of real-time information, these inputs are of little use, as there is little point in backing off a terminal for a time determined by some statistical analysis of fade duration or maintaining an outage state until some time has elapsed that is statistically relevant to inter-fade durations, when the precise fading conditions are known and updated via a suitable protocol several times every second.

Likewise, it has been found that the peak of fading events has little or no correlation with any of the first or second order dynamics [18], and provides no useful information whatsoever.

At the inception of this work, fade slope was thought to provide useful information, either as a method of short-term fade prediction, or in developing link budgets with unevenly spaced activation thresholds, designed with the help of fade slope statistics. However, as the work progressed it became obvious that the short-term fade predictors based on fade slope were inferior to adaptive signal processing techniques (see chapter 6), and that the activation thresholds were fixed as a property of the coding scheme used within the link budgets, without any possibility of customisation.

For the reasons above, the fade controller designed in this work uses only the short-term fade predictor as an input, although an investigation of second-order fade dynamics at V band was also undertaken and appears in the next chapter.

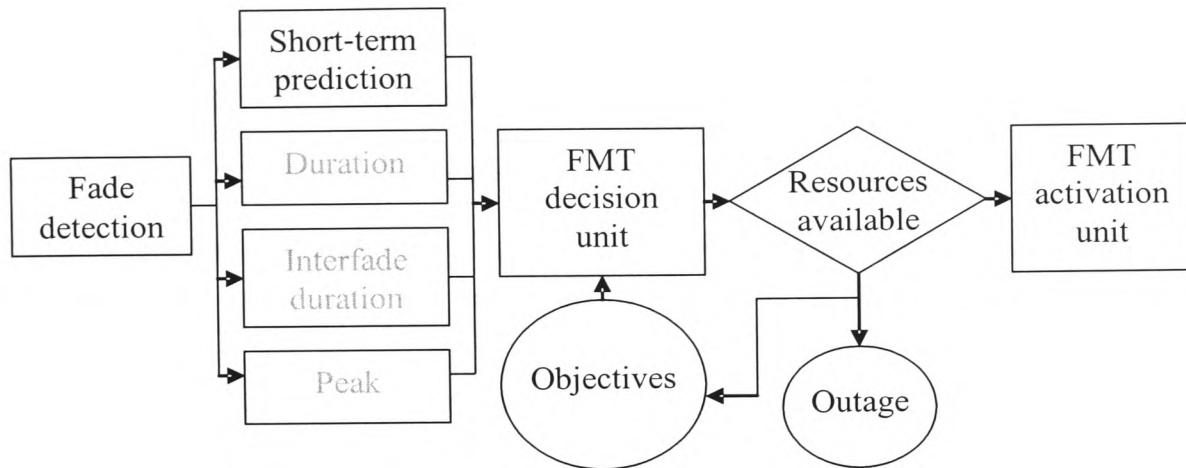


Figure 2.8: A generic fade controller

2.3.3 Fade Detection

There are several methods of estimating fading, each with certain advantages and disadvantages. These are described in detail in [53] and summarised below in order to justify the eventual choice.

The first method is to measure the C/N ratio of an unmodulated carrier at a frequency close to that of the downlink. This is the same method in which most satellite data gathering is performed and has the major advantage of being the most accurate of all the methods. The major disadvantage is that it would require all commercial satellites to carry beacons at a particular frequency, which could only be realised effectively with a beacon in each transponder. A real-time method of clear-sky estimation is required such as the approach taken in [63] and a frequency scaling method is also required to estimate the uplink attenuation.

A second approach is to estimate the C/N either from known data at the receiver, or from the unknown information-bearing portion of the signal [52]. Of these, the known training sequences perform better. These estimators have the advantage of not requiring

Chapter 2 – Fade Mitigation and Control

any component that does not already exist in current satellite network, except for the estimator itself, but at the cost of a reduced accuracy in comparison to beacon measurements. They are also only capable of measuring the overall link attenuation, although uplink attenuation at the hub is still detectable as discussed earlier.

A third method involves making use of the bit error rate to detect fades rather than the C/N ratio [64]. This also does not require any extra components but requires a lot of data in order to make accurate predictions. This makes the method extremely slow, which is a large drawback considering that it is the current event dynamics that are of interest.

Lastly, meteorological data and satellite imagery can be interpreted to determine the attenuation per locality rather than per individual link [60, 65]. Estimation by this technique cannot produce accurate estimates of attenuation but could be used by a network control algorithm at the hub to assign resources to regions that are likely to be experiencing fading.

Of these methods of fade detection, beacon measurements are most desirable for per-link analysis, with estimates from training sequences being the most desirable in the absence of beacon signals. However, due to the possible limitation of accurate beacon measurements from commercial satellites, C/N estimation by a known bit sequence has been chosen.

2.3.4 Short-term Fade Prediction

Short-term fade prediction is an important component of any FMT controller where the fade detection method involves estimation of the C/N ratio, the main aim being to reduce the prediction error as much as possible so that the detection margin required is also reduced.

Various methods of predicting future values of attenuation based on n previous samples have been proposed. These include fade slope methods [66, 67], adaptive signal processing methods [21, 68], and statistical methods [69]. These methods are described more thoroughly in chapter 6, along with results for their comparative performance.

2.3.5 Resource Allocation

In a centralised allocation scheme, the hub is charged with the task of assigning resources to individual users, its aim being to maximize QoS both in terms of adapting resources to account for traffic loads and adapting the communications parameters for individual links to account for signal fading. A large amount of work has been done in this area at Ka band including methods of estimating throughput of adaptive TDMA systems [70], analysis of resource allocation for fade mitigation [40, 41, 71], and much work on the combination of resource allocation for fade mitigation and traffic considerations [72, 73].

The resource allocation has to be carried out by a controller that uses parameters such as the current fading level, the short-term prediction of the fading level and scintillation variance. Controllers that perform this function have been proposed, such as by the use of Markov chains [28], and neural networks [29], but could also be implemented as an application-specific software algorithm.

There is also the question of the VSATs, as these require a controller that is able to request resources based on fade conditions, but is not involved with optimisation of traffic unless a distributed scheme is employed.

A simple resource allocation procedure was chosen for this work that is the same as the optimised-centralised policy in [74], where it is assumed that a centralised control strategy is taken.

2.3.6 Protocol Design

Design of a protocol for efficient communication via satellite depends on what the protocol is intended for. Several centralised and distributed protocols have been proposed for *Ka* band communications [13, 75, 76], and the DVB-RCS standard [11] has a particularly complex protocol that allows many types of traffic with different methods of allocating resources and even two types of data encapsulation.

However, these protocols do not take into consideration the increased fading levels and hence the drastic reduction in availability at *V* band. For this reason, a protocol has been designed using the above sources for inspiration. The protocol provides a starting point from which the links can be designed, and also introduces the idea of a separate control channel. This dual-channel protocol is discussed in detail in chapter 7.

Chapter 3

Second Order Fade Dynamics at V band

3.1 Introduction

Atmospheric fading at Ka and V bands has been studied extensively [77, 78] and broad agreement has been reached that satellite systems operating at these frequencies will no longer be able to rely on fixed attenuation margins to provide availabilities approaching those which the current fixed margin C and Ku band services provide. One of the major aspects of interest in the attempt to find an efficient and reliable solution has been the dynamics of rain-induced fades.

Three major dynamics of rain attenuation are of interest; fade slope, fade duration and frequency scaling. Fade slope is the time derivative of rain attenuation and is important in determining the required switching speed of fade mitigation techniques. It has also been used in several ways to make short-term predictions of attenuation. Fade duration can be sub-divided into two further categories; fade duration and inter-fade duration. Fade duration indicates the length of time that a fade exceeds a threshold value and inter-fade duration indicates the length of time that a fade drops below a certain threshold within an event. These concepts are illustrated in figure 3.1 (reproduced from [16]).

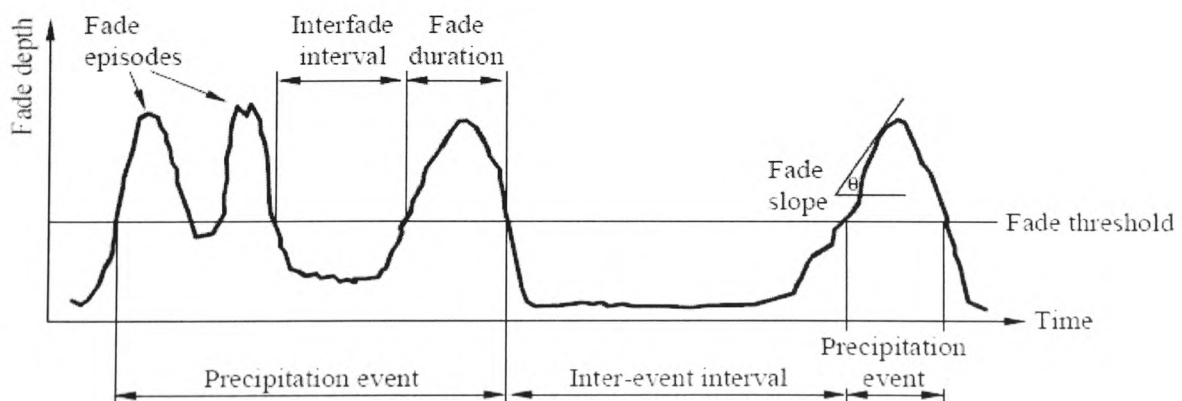


Figure 3.1: Fade Slope and Fade Duration

Frequency scaling of attenuation allows the attenuation at one frequency to be estimated by measuring the attenuation at another. This is particularly useful for FMTs such as uplink power control where uplink beacon attenuation can be estimated from downlink beacon attenuation.

While the first order dynamics of the particular data set that was used during this work has already been studied exhaustively [77], and fade duration has been studied to a lesser extent [79], fade slope had not previously been analysed. This chapter aims to review the fade slope and fade duration results from the data set provided by RAL.

It is noteworthy that with the exception of frequency scaling, the fade dynamics studied in this section were eventually not useful to the fade controller envisaged. However, the study was undertaken first chronologically, with only the generic fade controller of figure 2.7 as a guide for the work that was necessary.

3.2 *Fade Slope*

Much analysis has been carried out on the fade slope of rain attenuation. Earlier studies began as far back as the 1970s to 1980s when 12 GHz beacon signals from the SIRIO and OTS satellites were analysed in terms of their fade slope [80, 81]. It was found that filtering of scintillation changed the fade slope dynamics and this effect was characterised in [82].

Later experiments involved the Olympus satellite, which carried beacons at 12, 20 and 30 GHz and the ACTS satellite, which carried beacons at 20 and 27 GHz.

Measurements from various locations were taken from these satellites including sites at Blacksburg-Virginia, U.S. [83, 84], Eindhoven, Netherlands [85], Sparsholt, UK [86] and Fairbanks-Alaska, Tampa-Florida and White Sands-New Mexico in the U.S. [87]. The results from these studies were used by van de Kamp, along with the results in [82] to develop a distribution model of fade slope which is dependent on attenuation level, filter bandwidth and on the time interval used for slope calculation [88]. This model has subsequently been adopted by the ITU-R under their recommendation on fade dynamics [89]. The model itself appears in appendix A for reference.

The introduction of V band beacons on the ITALSAT satellite has allowed verification of this model at higher frequencies, which is necessary given the model's assumption of constant ratio of equiprobable attenuations between frequencies. However, although further studies on fade slope have been carried out at sites including Sparsholt, UK [90], Spino d'Adda [91] and Munich, Germany [92], the only study that has evaluated the model at these frequencies is that done with data recorded at Barcelona [93].

If a fade slope model is to be used at all at V band, it is important that the performance of the model is first evaluated to determine whether it is suitable, especially considering the reduced accuracy of the previous assumption at frequencies above 30 GHz. Figure 3.2 shows that for increasing attenuation levels, the scaling factor can be considered constant over a range of attenuations only when frequencies are similar. For example, an estimate of the attenuation at 12.5 GHz made from 7 GHz signal would require a scaling factor of approximately 3 regardless of the attenuation but to estimate the attenuation of a 50 GHz signal a scaling factor of 32 at 1 dB but only 15 at 15 dB would be required.

Consequently, *V* band frequencies may not adhere to the proportionality of the model as accurately as lower frequencies. However, since higher values of attenuation are more important to *V* band system design as they lead to a greater probability of large values of fade slope, the straight line approximation remains useful. A more accurate model would require not only the incorporation of this frequency scaling non-uniformity, but other parameters affecting the constant s , which determines the standard deviation of the model (see appendix A), such as the elevation angle and convectivity of rain, amongst others [88], as well as an improved probability distribution.

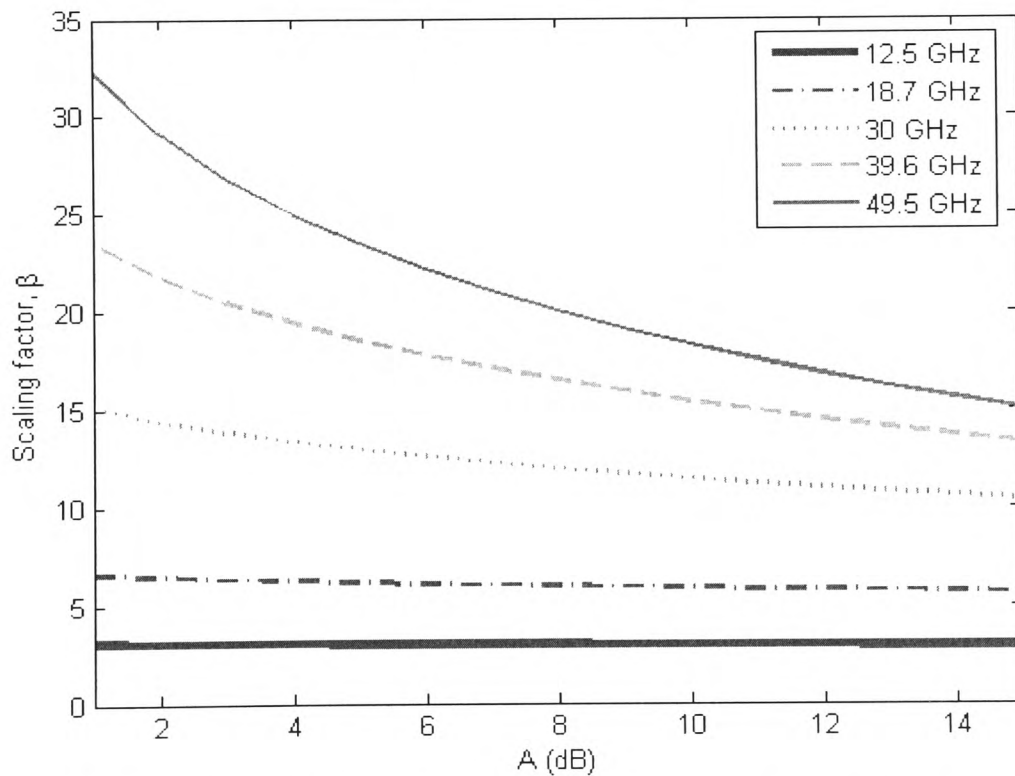


Figure 3.2: Frequency scaling of attenuation of various frequencies from 7 GHz

Data from Sparsholt, UK for two separate years was chosen, these being September 1997 to August 1998 and August 1999 to July 2000. The time series received had been pre-processed by the method described in [77] to determine clear-sky level. However,

the excess attenuation time series contained attenuation due to scintillation, which had to be removed before the fade slope of rain attenuation could be studied. This was achieved with a frequency domain filtering method similar to that described in [94] as the results were found to be more accurate than those found by time domain methods. A frequency of 20 mHz was found as a lower limit for scintillation in [95], and this was used as the cut-off frequency of the low-pass filter to ensure that all scintillation components were removed.

The filtering method chosen was that of a sharp cut-off frequency and was implemented by using a Fast Fourier Transform (FFT). The time domain signal was transformed to its corresponding spectrum, all components above the cut-off frequency were removed and an inverse FFT was used to transform back to the time domain. Time series end effects were reduced by removing 11 samples from the beginning and end of every block of filtered data. The number of samples to remove was determined using the relationship between moving average and FFT filters found in [88].

It is noteworthy that Baxter *et al* have analysed offline fade slope filtering methods and found that wavelet domain filters are optimum [96].

The number of data points used for each beacon frequency in the fade slope analysis is shown in figure 3.3. Lower attenuation depths have a greater number of samples as do higher frequencies, with the exception of the 40 and 50 GHz beacons at attenuations above 15 dB, where the 40 GHz data has more samples than the 50 GHz data. This tendency can be attributed to the fact that different attenuations are caused by different atmospheric conditions for the three frequencies used. For example, if a common type

of convective event causes a peak attenuation of 16 dB at 40 GHz, a peak attenuation of 20.5 dB could be expected at 50 GHz and would therefore not be recorded in the results, since the maximum dynamic range of the measuring equipment was 20 dB. Also, not all useable samples at one frequency coincide with useable samples at other frequencies, meaning certain events are counted at one frequency but not at another. This could be another factor.

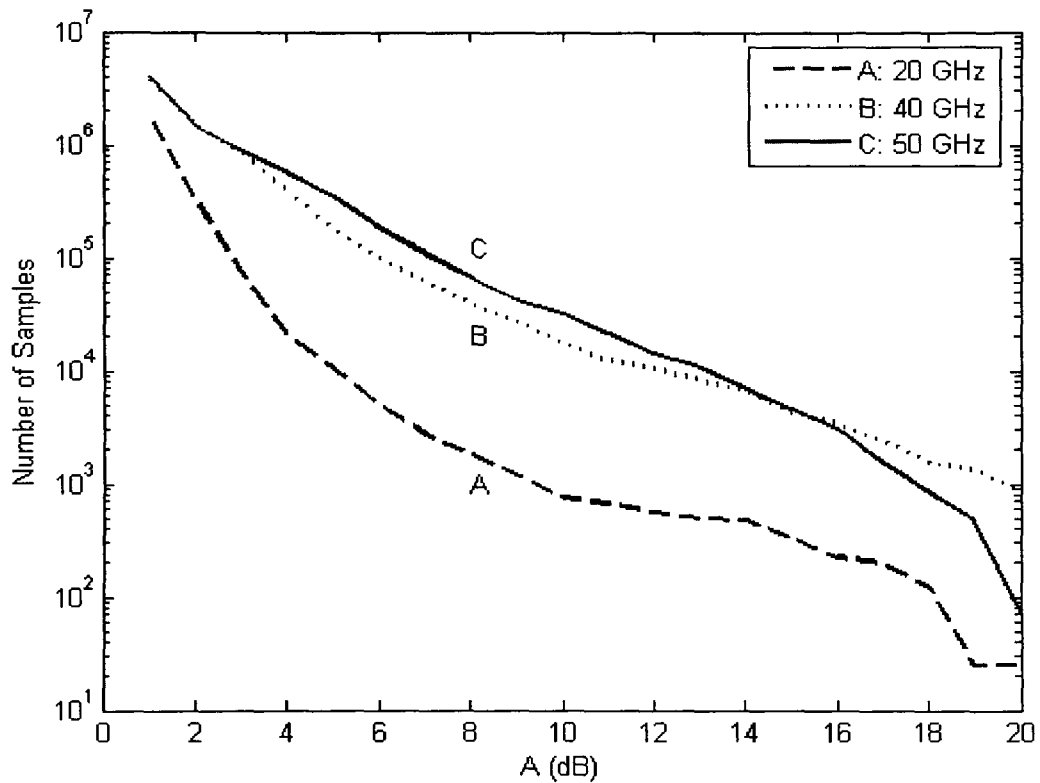


Figure 3.3: Number of samples used to generate fade slope statistics from 1 to 20 dB at 20, 40 and 50 GHz

Joint statistics of fade slope ζ and attenuation A were generated by storing fade slope values in bins of size 0.001 dB/s and 1dB for ζ and A respectively. A weighted mean of the fade slope standard deviation for all three frequencies at all attenuation values was calculated, as there was more data at higher frequencies, due to the frequency dependence of attenuation. A filter bandwidth of 20 mHz and sampling interval of 1

second were used in this calculation. The fade slope standard deviation for all three frequencies and the weighted mean are shown in figure 3.4. This analysis was restricted to attenuation levels not exceeding 17 dB even though statistics were generated up to 20 dB, since the data was too sparse to give reliable results above 17 dB.

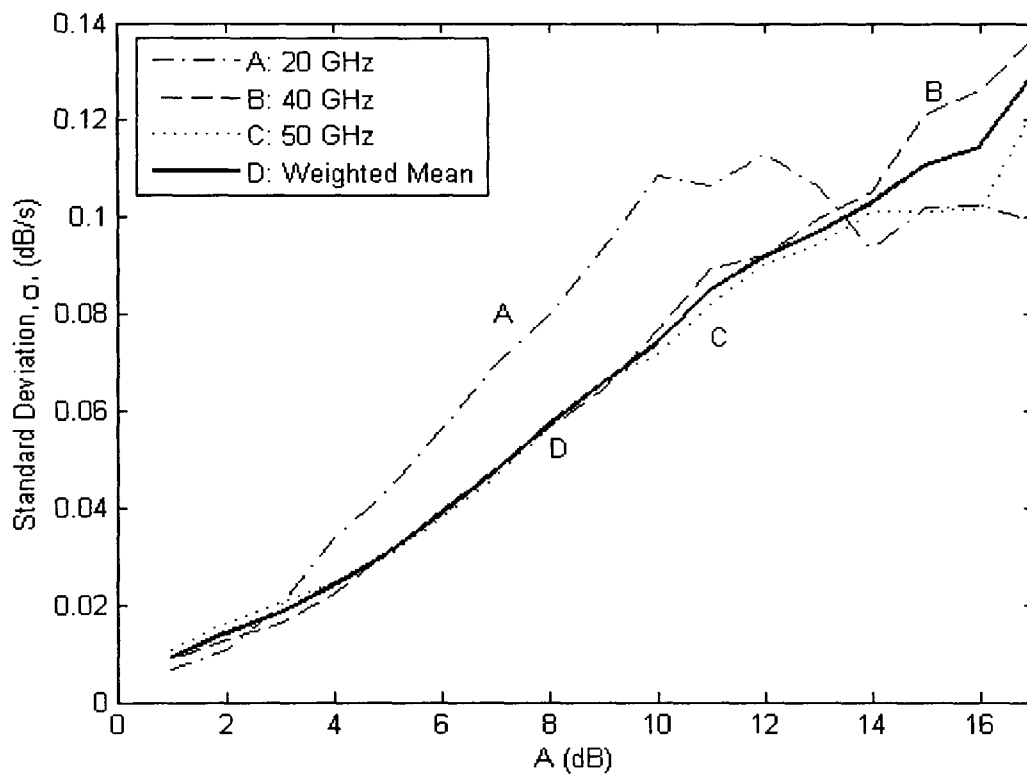


Figure 3.4: Standard deviation of fade slope

The zero offset best fit to the weighted mean was found and equation (A.1) (see appendix A) rearranged to find s from the known parameters. The resulting value of s was found to be 0.0121, where s is a scaling factor which depends on the climate and elevation angle as defined in appendix A.

The 20 GHz results deviate quite significantly from the 40 and 50 GHz results in the mid range of attenuation values, being noticeably larger. One possible reason for this is the varying degree by which the beacon signals are affected by various atmospheric

processes. For example, events that are likely to cause attenuations of 4 to 12 dB at 40 and 50 GHz are clouds and light/stratiform rain. These events do not change very rapidly, resulting in low fade slopes. However, attenuations of 10 dB for 20 GHz signals are the result of quite heavy convective rain, which changes rapidly and results in higher fade slopes.

From visual inspection of the weighted mean, there appears to be some slight curvature at low values of attenuation, indicating that the straight-line approximation may not be entirely accurate. This can be seen more clearly in figure 3.5, where s has been computed independently for each value of A at each frequency from equation (A.1). The point at which the results begin to curve appears to depend on frequency, higher frequencies beginning to curve at higher values of A . This could also be due to the effect of cloud attenuation on the results for 40 and 50 GHz.

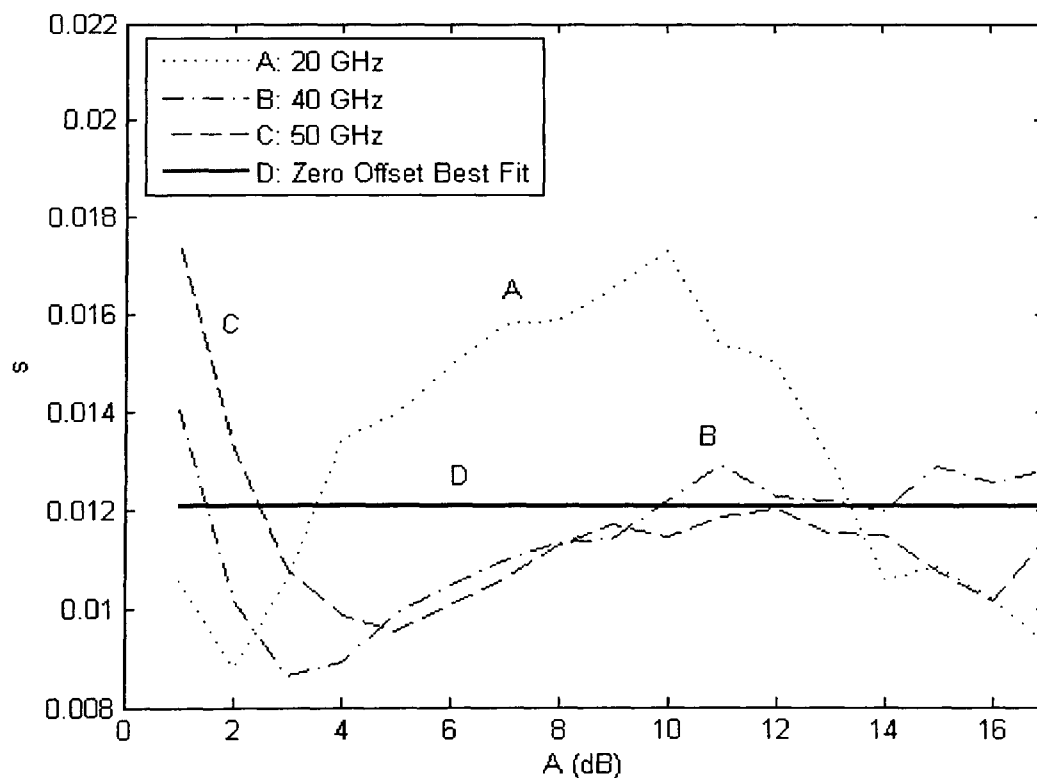


Figure 3.5: s parameter calculated for each attenuation threshold and the zero-offset best fit.

A second order polynomial was fitted to the weighted mean standard deviation but resulted in an extremely low negative value for the second order coefficient, leaving a curve that is visually indistinguishable from a straight line. Higher order polynomials were also fitted to the weighted mean, but do not begin to reduce the squared error significantly until the fourth order, indicating that a more accurate model would sacrifice simplicity for relatively little improvement. This does however leave some apparent discrepancy in the conditional PDFs of fade slope at low values of attenuation and is broadly in agreement with the results found from Barcelona and Milan in [93].

Figure 3.6 shows the conditional PDF for attenuation levels of 1 to 4 dB against the corresponding PDFs from the ITU-R model for the 50 GHz time series. Similar results were found at 20 and 40 GHz. The smooth curves represent the model and the jagged lines represent the results. The fit was analyzed in terms of the chi-squared goodness of fit and was found to fail this test for all confidence intervals. However, this can be expected for an empirical model. For more details on this test, see [18]. Figure 3.7 shows the CDFs for the 50 GHz time series at attenuation levels of 1, 4 and 15. The model parameter s was set to 0.0121 as was found from the data as opposed to 0.01 as recommended in [97]. For an in-depth analysis of the parameter s , see [88]. Again, similar results were found at 20 and 40 GHz.

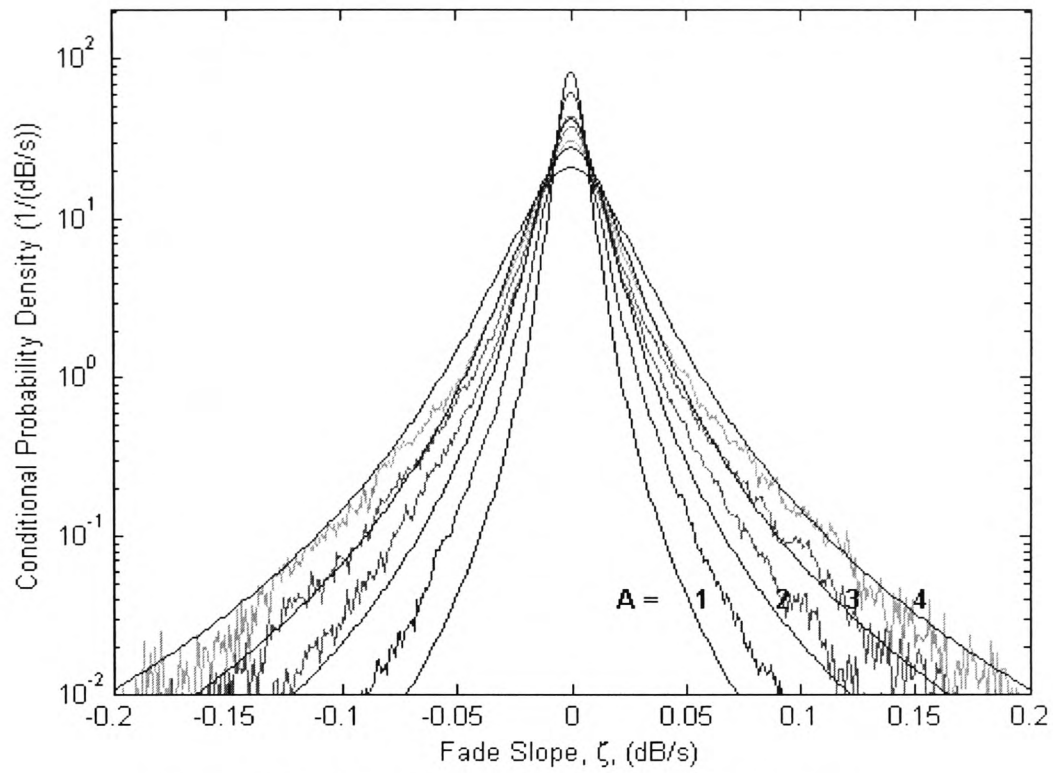


Figure 3.6: PDF of fade slope for 50 GHz time series

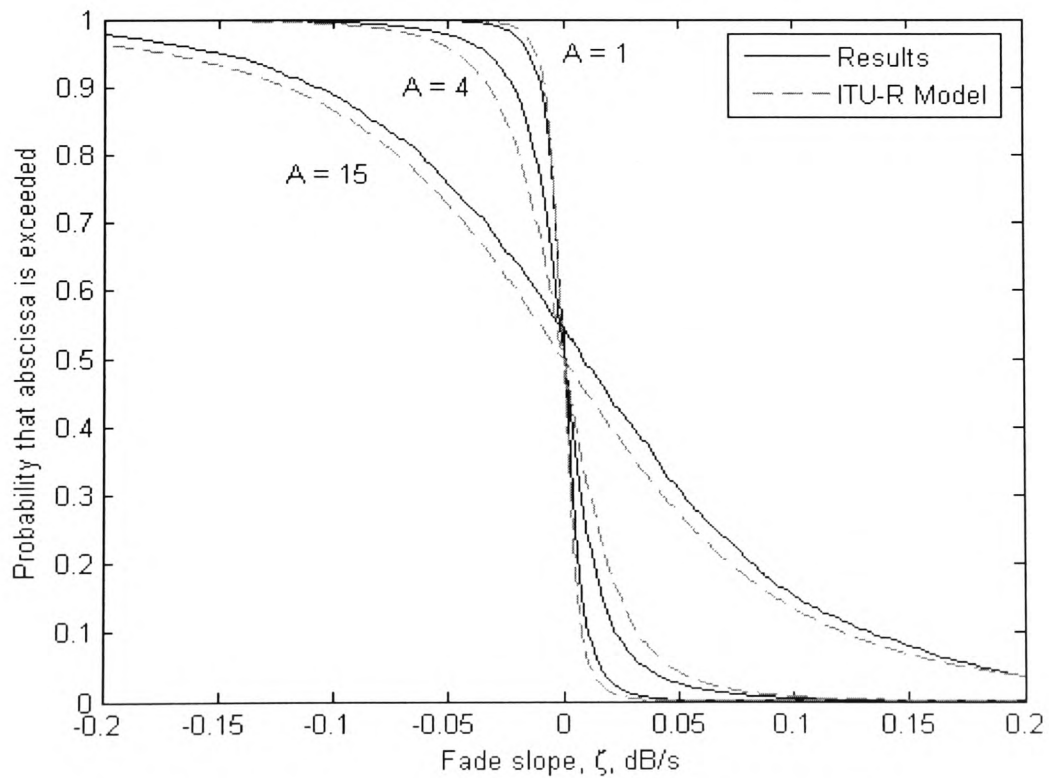


Figure 3.7: CDF of fade slope for 50 GHz time series

Chapter 3 – Second Order Fade Dynamics at V band

In order to examine slope symmetry, the cumulative distributions of fade rise rate and fade fall rate were calculated from the PDF and are shown in figure 3.8 for the 50 GHz time series.

While the fade rise rate and fade fall rate appear symmetrical in the PDF, the CDF reveals that there is a departure between the two at higher values of fade slope, especially for higher attenuations which are likely to be caused by convective storms. The data for these values of fade slope is sparse and therefore less reliable than lower values of fade slope, but can be attributed to heavy thunderstorm activity where the rain rates preceding or following the convective cell are different, the difference depending on the ratio of leading to trailing stratiform events [98]. Similar results were also found in [99] and [92]. This effect was also seen for the 20 and 40 GHz time series, although the departure between fade rise rate and fade fall rate tended to be smaller at lower frequencies.

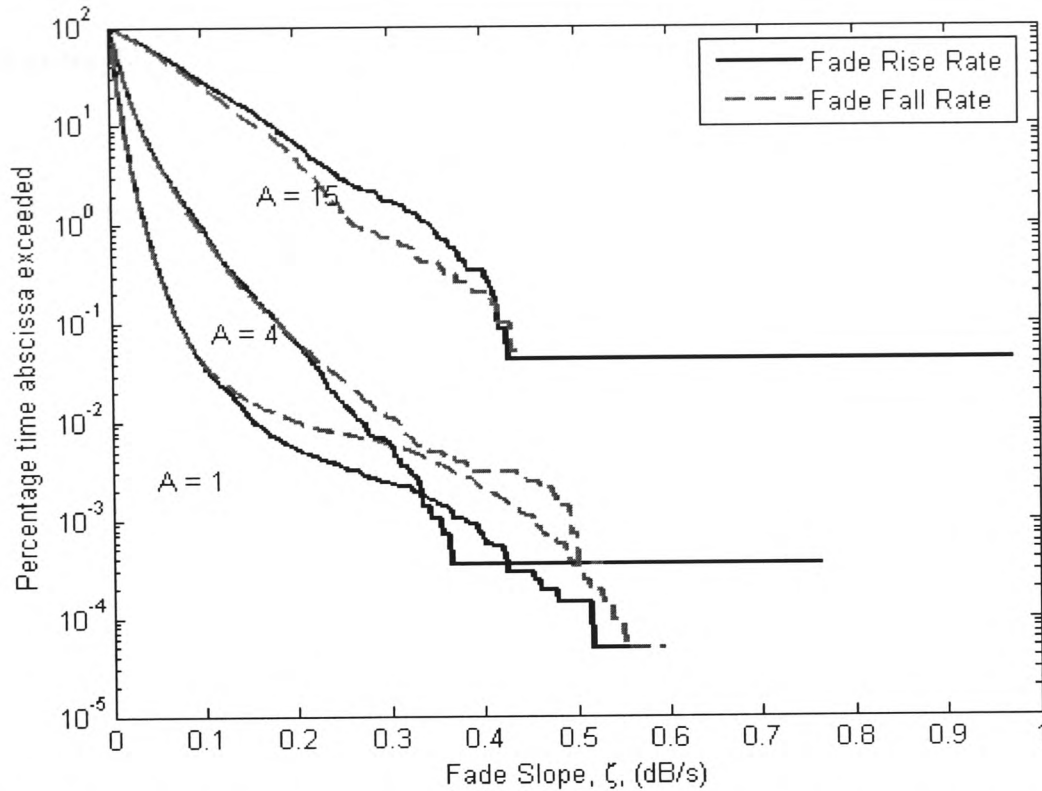


Figure 3.8: CDF of fade rise rate and fade fall rate at various levels of attenuation

Fade slope for the periods October 1997 to March 1998 and August 1999 to July 2000 was also considered on an hourly basis in order to analyze any diurnal trends. The results for the 50 GHz time series is shown in figure 3.9, with similar results being obtained at 40 GHz, although the 20 GHz beacon data was too sparse to provide reliable results. The results show a slight tendency for higher fade slopes at midday and midnight. However, the rise at midnight could be the product of inaccuracy in the measurement of fade slope at the start and end of each day. This arises from a lack of samples surrounding the end points and is a source of inaccuracy for time or frequency domain operations alike. The effect was reduced by removing some samples from either end of the filtered time series, but the number of samples used was calculated from an empirical observation of the relationship between cut-off frequency and the size of a

moving average filter in [88], therefore some inaccuracy may remain at the start and end of every day.

Fade slope for the period August 1999 to July 2000 was measured on a seasonal basis in order to analyze any seasonal trends. The results appear in figure 3.10 and show a definite seasonal dependence, with autumn in particular tending to have higher fade slopes. This is largely due to the seasonal variation of weather, convective rainstorms occurring more frequently in southern England in summer and autumn [100]. It is also noteworthy that stronger winds are a feature of the autumn season in Southern England. The same trend was also found from the 20 and 40 GHz data.

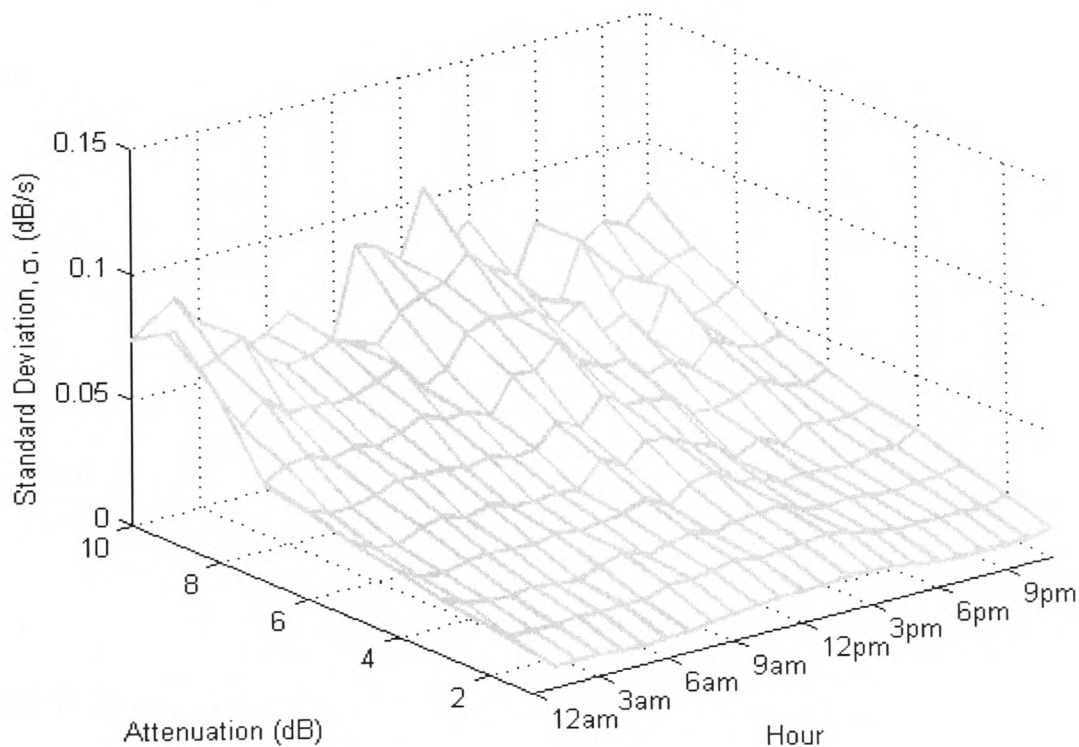


Figure 3.9: Diurnal variation of fade slope standard deviation and attenuation at 50 GHz

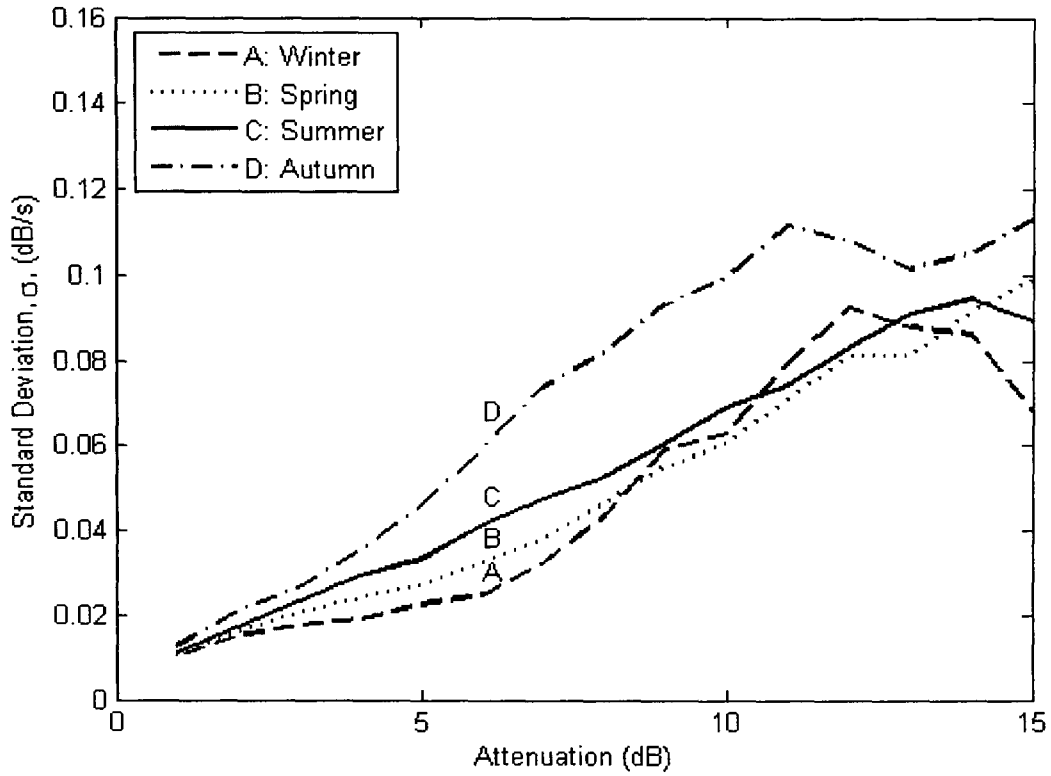


Figure 3.10: Seasonal variation of fade slope standard deviation and attenuation at 50 GHz

For further information, including correlation analysis and a more in depth discussion of the pre-processing and filtering, see [18].

3.3 Fade Duration

A subset of data from the same dataset as the fade slope analysis was analysed between August 1999 and July 2000. Joint statistics of fade duration, D and attenuation, A were generated in bins of size 1 second and 1 dB for D and A respectively. An event was defined as being between two 1 dB threshold crossing points, the first crossing having a positive slope and the second having a negative one. Within an event, the duration of all threshold crossings from 1 dB to 20 dB were recorded as separate histograms. Inter-fade durations for the same period were also recorded in histograms of the same dimensions

as those used for the fade duration statistics, where an interfade event is defined as being between two threshold crossing points within an event, the first crossing having a negative slope and the second having a positive slope. Since an event was defined as starting where the 1 dB threshold is crossed, inter-fades were recorded for attenuations ranging from 2 dB to 20 dB.

There are two fundamental problems to overcome when calculating fade duration from beacon measurements and a third that arises when measuring diurnal variability:

1. Preventing random noise from causing spurious threshold crossings
2. Accounting for invalid data that occurs within an event
3. Establishing the most appropriate time of day to bin an event

The first problem was overcome by the use of a small amount of hysteresis. For a threshold crossing with a positive slope to be recorded, the attenuation had to exceed the value of the threshold plus a certain margin of hysteresis. Similarly, for a threshold crossing with a negative slope to be recorded, the attenuation had to fall below the value of the threshold minus a certain margin of hysteresis. However, setting the margin too high could remove threshold crossings due to scintillation, and for this reason a small hysteresis margin of 0.1 dB was used.

The second problem of invalid data can cause spurious statistics when one considers that a single sample with an associated data flag of 0, i.e. bad data, can cause one event to be split into two shorter events. This problem is compounded by the fact that long events are more likely to have bad data than short events and therefore, according to the results, the longer events would have an unrepresentatively low probability of

occurrence, while short events would have an unrepresentatively high probability of occurrence. However, if large sections of bad data are interpolated they are not representative of short threshold crossings due to scintillation and also increase the chance of over-estimating event duration, especially when the invalid samples occur near the start or end of an event. The duration of invalid portions of data was analysed in order to find an appropriate upper limit for interpolation length. The cumulative distribution of invalid data duration is shown in figure 3.11.

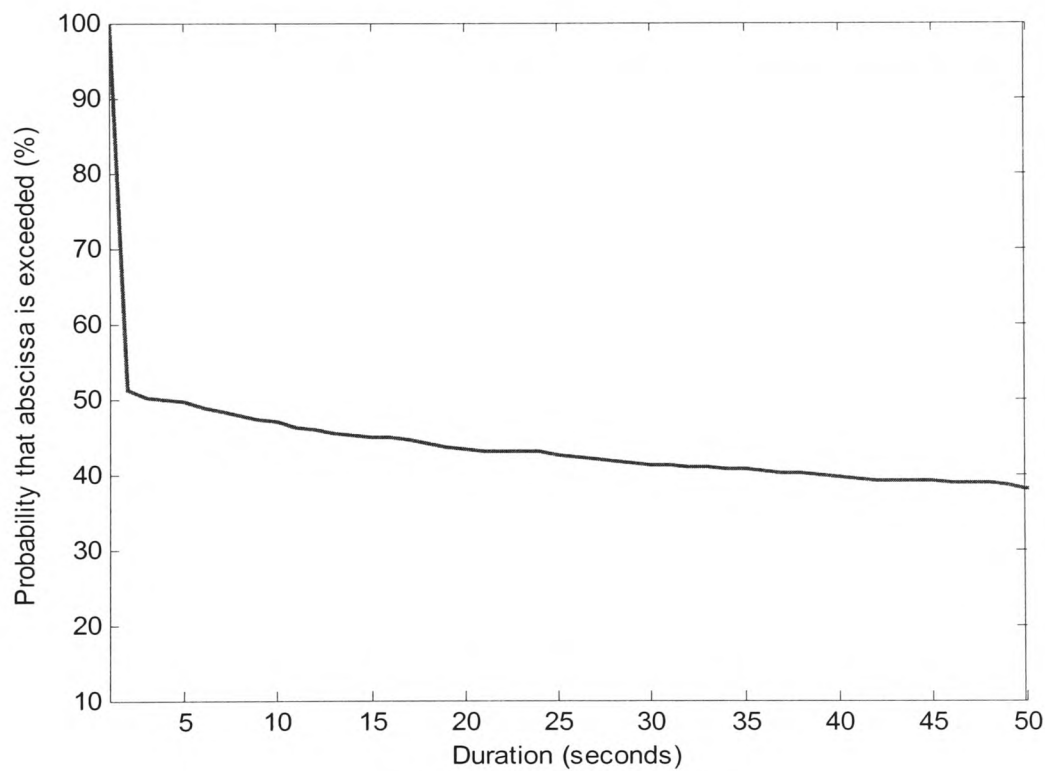


Figure 3.11: CDF of invalid data durations

It is clear that the majority of durations of invalid data are quite short, and for this reason a maximum interpolation length of 10 seconds was used, as this removes 53% of the total number of invalid periods, while only resulting in a number of interpolated samples equal to 0.003% of the total samples. It is also short in comparison to the

duration of rain events, and therefore is unlikely to over-estimate events that have invalid data close to the beginning or end.

The third problem arises from the fact that the data is stored in individual days, and has a period of about 60 seconds of invalid data at the beginning of every day. For this reason, the days could not be evaluated continuously but had to be truncated, i.e. events were ended at the end of every day. This not only added to the previous problem of splitting events, but could also bias the statistics depending on where the event was binned on a diurnal basis. For example, if the event was binned at its beginning, then the first diurnal period will be positively biased, whereas if it is binned at its end, the last period will be positively biased. To reduce this effect, a diurnal period of 4 hours rather than 1 hour was chosen to accumulate more samples per period, therefore reducing the impact of any bias. Also, the diurnal period containing the mid-point of the event was used rather than the period containing the starting point of the event.

The results for the probability of occurrence of a fade event above 1 dB for the 50 GHz beacon is shown in figure 3.12. The figure also shows the results from the ITU-R model for fade duration [89], which is based on the work in [101].

The model over-predicts the fade duration by as much as an order of magnitude in some places. The results for the 20 and 40 GHz time series are similar, the model significantly over-predicting fade duration for all exceedance probabilities. This could in part be caused by invalid data as discussed earlier; in addition, scintillation intensity scales with frequency, giving rise to a larger number of threshold crossings of short duration. Since invalid samples cannot be avoided, it is difficult to determine whether the difference

between the model and the results is due to the invalid samples or a shortcoming of the model itself.

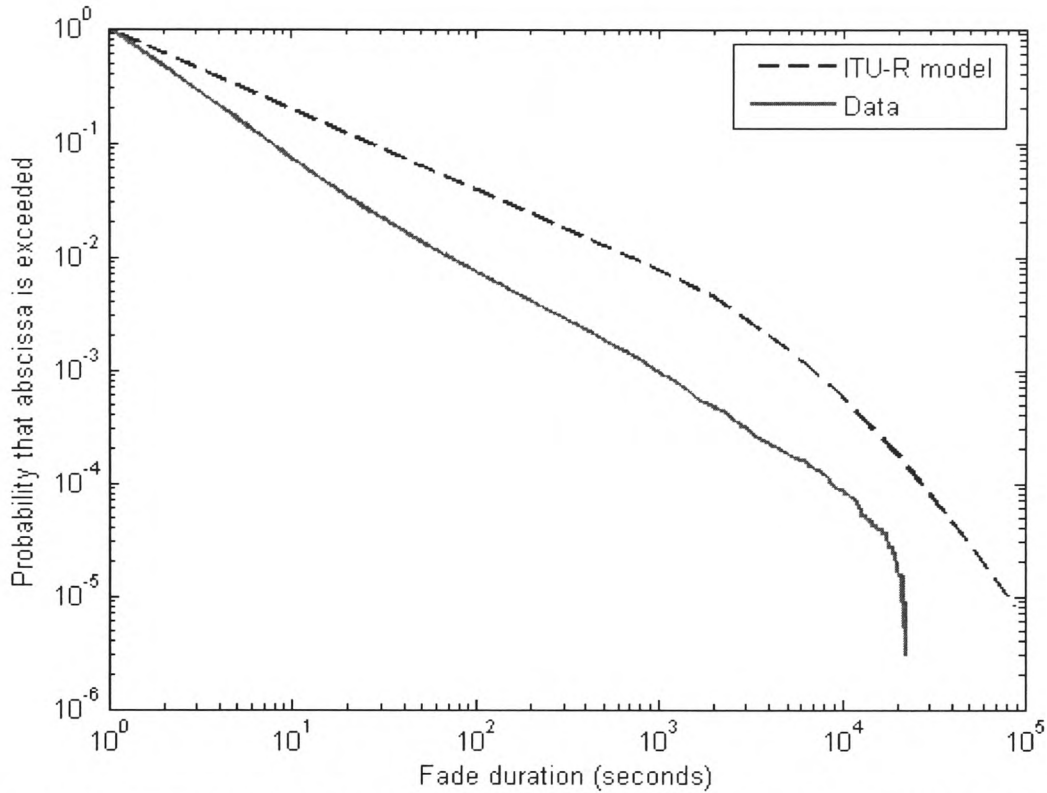


Figure 3.12: CDF showing the probability that a given duration is exceeded at the 1 dB threshold for the 50 GHz time series

The ITU-R model also allows the absolute number of events to be calculated if an optional parameter of the total fading time for a given attenuation is provided. While this can be calculated from a model CDF of attenuation such as [9], if experimental results are available, this is preferred. The results for attenuation levels of 1, 7 and 20 dB for the 50 GHz time series is shown in figure 3.13. Similar results were found for the 20 and 40 GHz time series.

These results show that the difference between the model and the results could be due to shorter durations and the power law part of the model since the lognormal model for

long durations seems to fit, whereas the model under-predicts the number of fades of short duration.

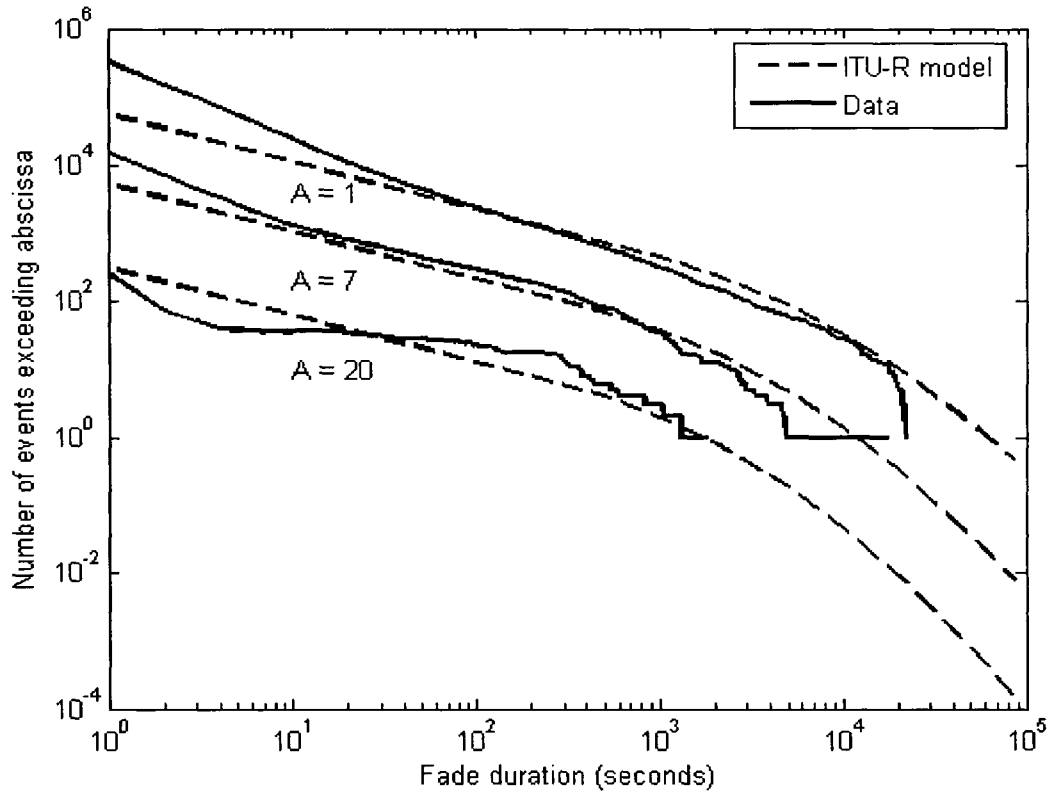


Figure 3.13: Number of events exceeding abscissa for the 50 GHz time series

The results for fraction of fade time exceeded due to fade events above the 1 dB threshold are shown in figure 3.14 for the 50 GHz time series. The model appears to fit fairly well visually for the 20 and 40 GHz time series, although it slightly over-predicts the probability of long fades for the 50 GHz time series. A graph of total exceedance time for the 50 GHz time series is shown in figure 3.15. Again the model appears to fit fairly well visually, with some variability for higher durations, which can be attributed to the low sample populations at these durations. The total cumulative durations in the 1-second bins of the various thresholds can be used to calculate the CDF of annual attenuation exceedance if required.

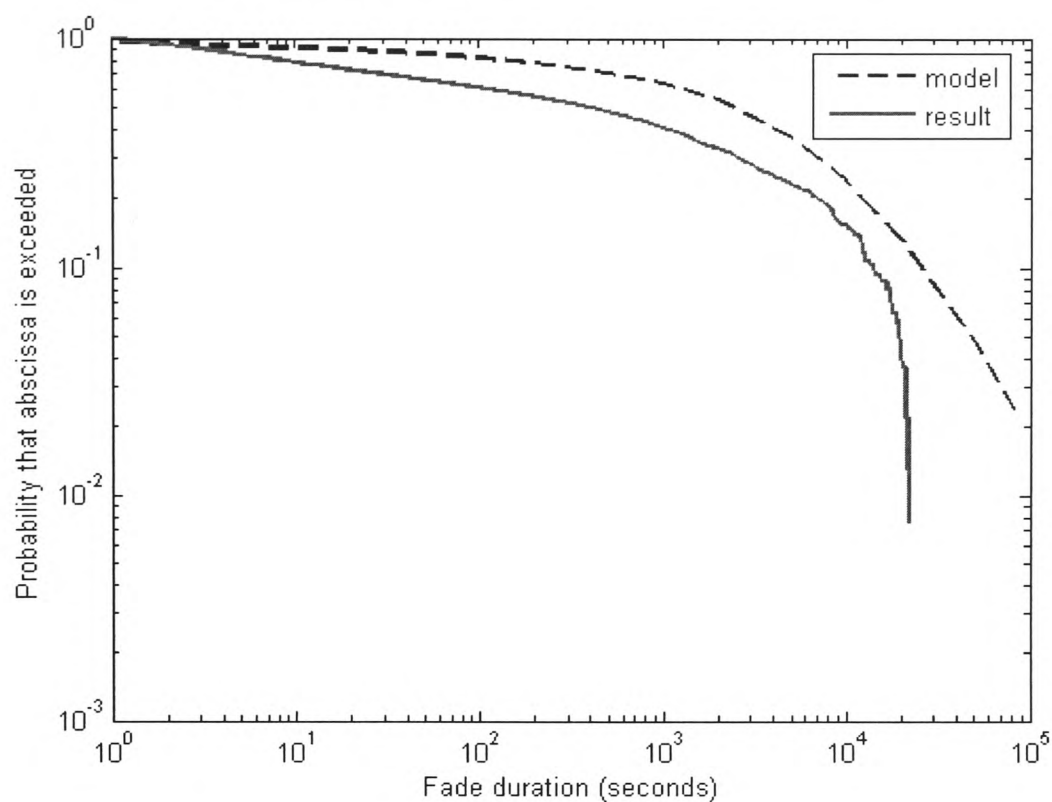


Figure 3.14: CDF showing the probability of total fade time exceeding abscissa at the 1 dB threshold for the 50 GHz time series

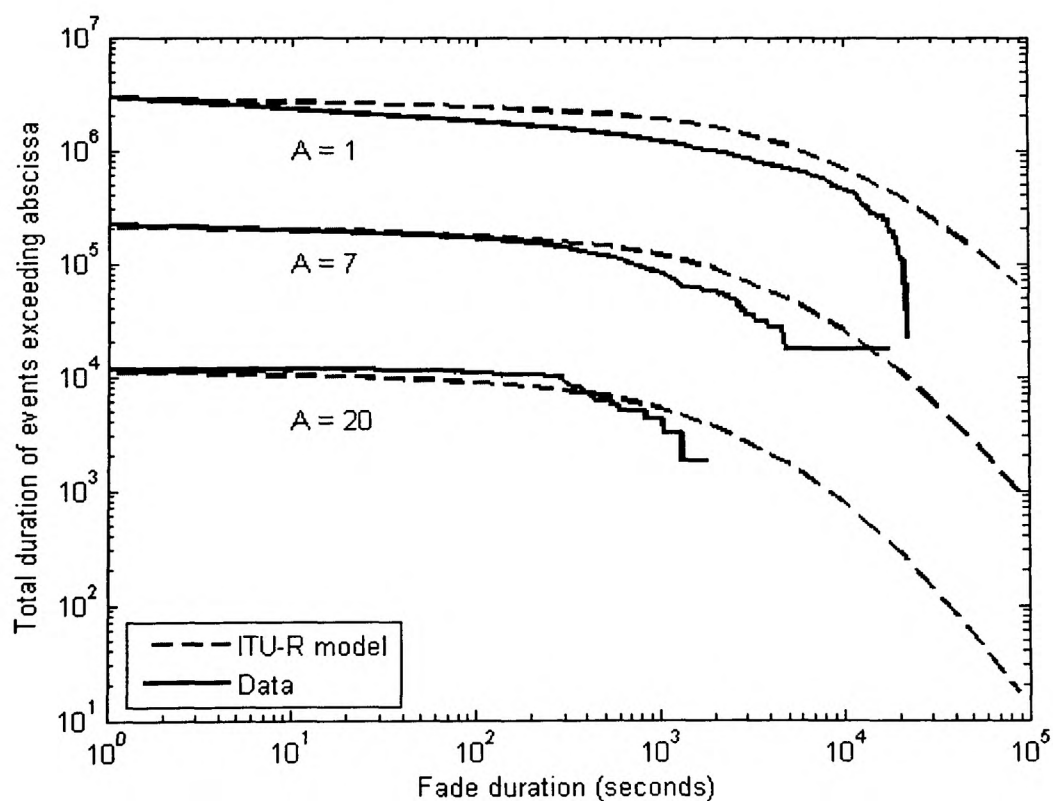


Figure 3.15: Total duration of events exceeding abscissa for the 50 GHz time series

It is useful to see how the model performs over the range of attenuations for various probabilities of exceedance. Graphs of duration as a function of attenuation for various probabilities of exceedance are shown in figure 3.16. The results show that the model over-predicts the duration for nearly all exceedance probabilities at nearly all attenuations. The recommendation itself also comments on the high variability of the accuracy of the model given different data sets [89].

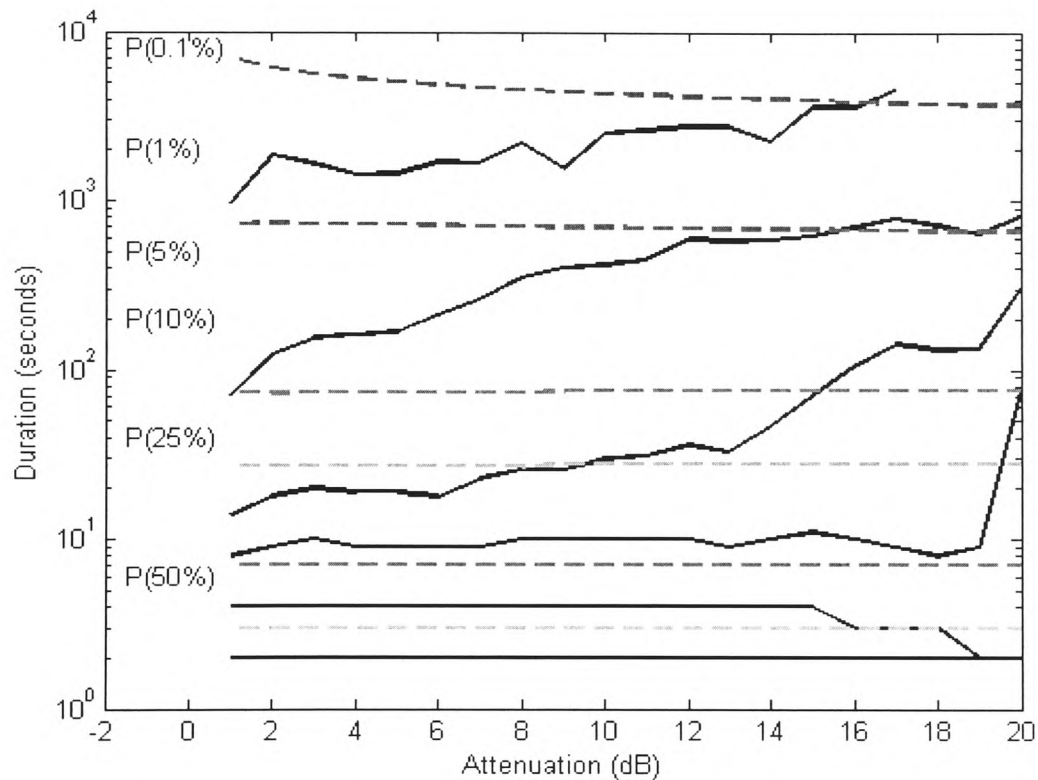


Figure 3.16: Fade duration as a function of attenuation for various exceedance probabilities for the 50 GHz time series

A diurnal and seasonal analysis of fade duration was also conducted but no trend could be reliably determined from the results due to the high variability of this parameter.

3.4 Inter-fade Duration

Inter-fade statistics were also recorded in terms of the probability of an inter-fade exceeding a specified duration, the total number of inter-fades exceeding a specified duration, the total fraction of inter-fade time due to inter-fades exceeding a specified duration and the total exceedance time of inter-fades of a specified duration. As yet there is no inter-fade duration model to compare the results with.

A graph of the number of inter-fades exceeding a specified duration is shown in figure 3.17 for the 50 GHz time series.

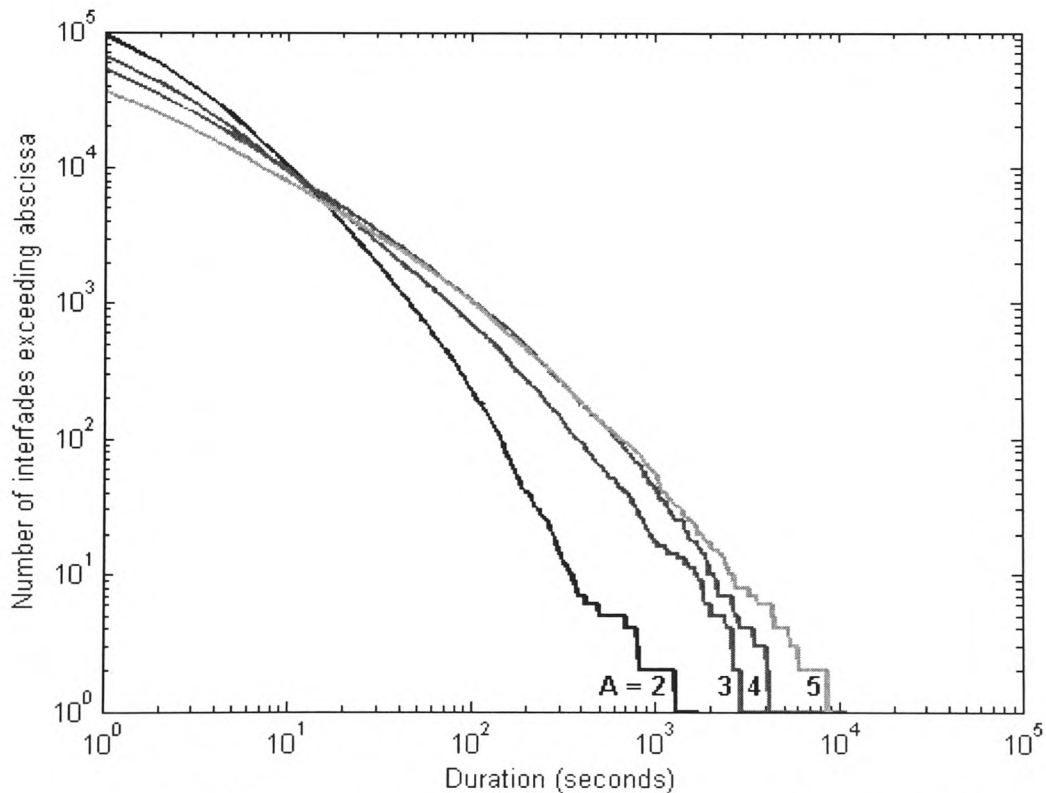


Figure 3.17: Number of inter-fades exceeding abscissa at 50 GHz

Similar results were found at 20 and 40 GHz, with the absolute number of inter-fades increasing with frequency and the point at which the graphs for each attenuation

threshold meet reducing with increasing frequency. Inter-fade duration is also dependent on the attenuation threshold, longer durations becoming more likely with higher attenuations. This behaviour can be attributed to the proportion of scintillation-induced threshold crossings to rain-induced threshold crossings. Scintillation alone can cause threshold crossings at lower attenuations but some form of rain event is required at higher attenuations. Therefore the ratio of rain-induced threshold crossings to scintillation-induced threshold crossings is higher at higher attenuations. However, as both rain attenuation and scintillation intensity increase with frequency, the ratio of rain-induced threshold crossings to scintillation-induced threshold crossings alters as a function of these two scaling factors. Since rain attenuation scales more significantly than scintillation, the effect is to increase the ratio of rain-induced threshold crossings to scintillation-induced threshold crossings and therefore reduce the point at which the graphs intersect.

Although the inter-fade duration was further analysed in terms of the total number of inter-fades exceeding a specified duration, the total fraction of inter-fade time due to inter-fades exceeding a specified duration and the total exceedance time of inter-fades of a specified duration, no additional insight was gained from these results and they have been omitted here for brevity. Likewise, the analysis of the effects of filtering on inter-fade duration and seasonal and diurnal variation gave no new information and these results have consequently been omitted.

3.5 *Frequency Scaling*

Frequency scaling of attenuation has a major application in the use of uplink power control. If a station monitors a beacon signal close to the downlink frequency, the attenuation on the uplink can be estimated and the transmitter power adjusted accordingly. However, within this work it has also been very useful in scaling attenuations for higher frequency beacons where loss-of-lock has occurred from beacon measurements at lower frequencies. It has also been useful for creating Ku band time series from beacon measurements at V band, the methodology of which is discussed in chapter 7.

Frequency scaling models have been proposed by several authors including Laster and Stutzman [102], Sweeney and Bostian [55] and Grémont and Filip [103] among others (see [104]). The ITU-R also has a model for calculating the instantaneous frequency-scaling factor [9]. This model has already been verified by several authors at V band frequencies, see for example [103].

The ITU-R model for frequency scaling was chosen and was used throughout this work.

3.6 *Summary*

This chapter has shown measurements of fade slope, fade duration and inter-fade duration using time series from RAL. Fade slope and fade duration have been compared with the ITU-R models, the fade slope model fitting the results to some degree at V band with some small error associated with scaling of equiprobable attenuations. This

Chapter 3 – Second Order Fade Dynamics at V band

gives some indication of how quickly rain fading occurs, which is a useful parameter as it guides future decisions of how quickly FMT schemes should switch.

However, fade duration results did not fit the ITU-R model well visually and no chi-squared goodness of fit test was required.

Frequency scaling of attenuation has also been briefly reviewed and the ITU-R scaling model chosen for use in the latter stages of work.

Chapter 4

Duo-binary Turbo Coding and Decoding

4.1 Introduction

In chapter 2 the various methods of fade mitigation were discussed and an overall scheme chosen that was suitable for the scenario of V band satellite Internet provision. Of these methods, the turbo code as defined for the return channel within the DVB-RCS standard [11] was chosen as the method of FEC coding. This particular code was selected for several reasons. The first was for the improved gain of the turbo-coding technique over older methods. The second was for the duo-binary nature of the code [105] which lends itself well to QPSK modulation and whose decoding converges more quickly on the corrected sequence than that of a binary code. The third was for the diverse range of code-rates already supported by the standard, allowing immediate flexibility with pre-defined puncturing sequences and lastly for the various block sizes supported by the standard, most notably that of a 53 byte block which corresponds exactly with the ATM packet size. Each of these blocks is also associated with an interleaving algorithm, optimized to give good randomness.

However, although the DVB-RCS standard defines the encoder, the puncturing sequence and the interleaving algorithm in detail, the subject of decoding is omitted altogether, perhaps because it is a more complex procedure. This chapter covers the turbo encoding process, as specified by the standard itself, a suitable decoding process developed in the course of this research which takes into account the duo-binary and tail-biting properties of the code, and simulation results of performance.

4.2 Turbo Encoding

4.2.1 The Encoding Process

The turbo encoding principle was introduced in [32], where two recursive systematic convolutional (RSC) codes in parallel with an interleaving stage were used to generate three coded output bits for every input bit. Since then, block codes and non-systematic convolutional codes have also been used with the turbo-coding principle [106, 107], although the code used in the DVB-RCS standard uses an RSC turbo code. Figure 4.1 shows a block diagram of the turbo encoding process as defined by the standard.

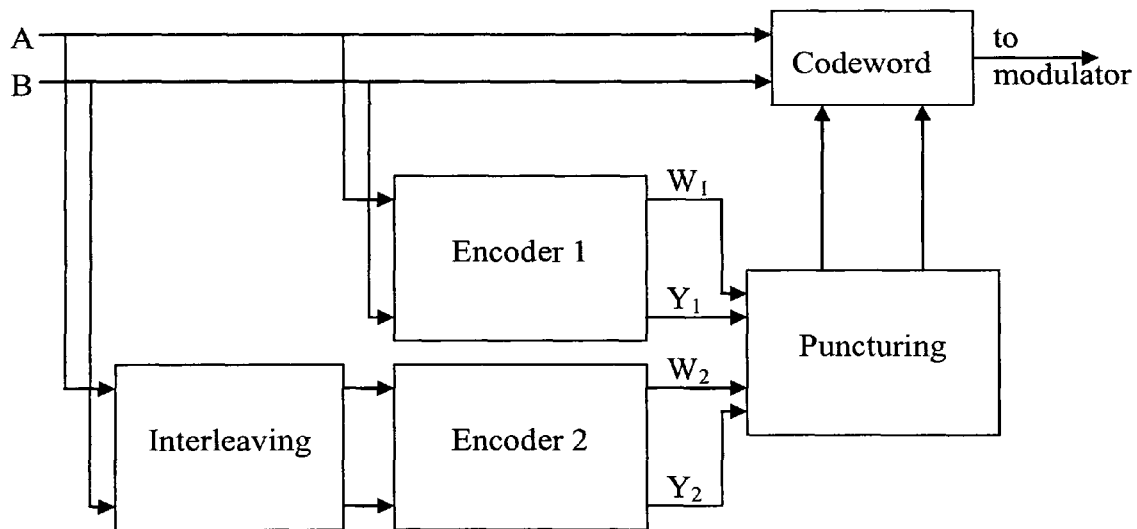


Figure 4.1: Block diagram of DVB-RCS Encoding process

Two consecutive bits are taken from a block of k bits, or N couples (where $N = k/2$), and these are added directly to the first recursive part of the codeword. The bits are also passed through an encoding stage sequentially, from which coded bits of W_1 and Y_1 are obtained, and, after some permutation at an interleaving stage, a second encoder produces coded bits W_2 and Y_2 . Although there is nothing preventing different encoders

from being used in the turbo-coding process, the same encoder is used within the DVB-RCS standard. The recursive and coded bits are combined with a puncturing algorithm defined by the code rate required and the final codeword is sent to the modulator.

The following subsections describe each of these components in more detail.

4.2.2 The Encoder

A convolutional encoder of constraint length 4 and a feedback branch is used as the component code for both the sequential and interleaved sequences. The symbolic notations for each of the branches are as follows:

- $1 + D + D^3$ for the feedback branch
- $1 + D^2 + D^3$ for the Y parity bits
- $1 + D^3$ for the W parity bits

This is more easily visible from the block diagram of the encoder in figure 4.2.

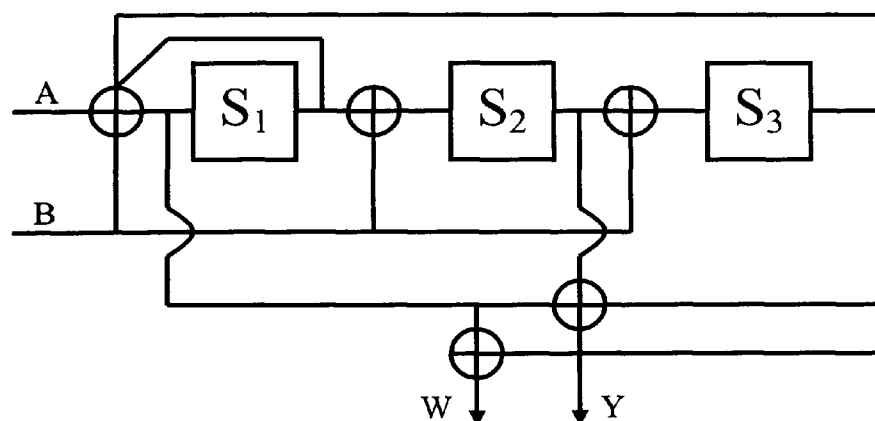


Figure 4.2: Component encoder for DVB-RCS turbo code

Chapter 4 – Duo-Binary Turbo Coding and Decoding

Each block of N couples must be encoded in two stages. This requirement arises from the circular or tail-biting nature of the code whereby the block is not terminated with zeros but instead the shift-register has the same initial state and final state. The first stage encodes the block with the encoder in the all zeros state and from the final state, a look-up table is used to determine the state in which the encoder must begin in order to terminate in the same state. The look-up table as defined in [11] is reproduced in table 4.1. The top row defines the octal state in which the shift-register terminates, the left hand column is the remainder after the number of couples in the block, N , has been divided by 7. The state S_c corresponding to these two values is the initial state that the encoder must be in order for the final state to be the same.

S_N^0	0	1	2	3	4	5	6	7
$N \bmod 7$								
1	$S_c = 0$	$S_c = 6$	$S_c = 4$	$S_c = 2$	$S_c = 7$	$S_c = 1$	$S_c = 3$	$S_c = 5$
2	$S_c = 0$	$S_c = 3$	$S_c = 7$	$S_c = 4$	$S_c = 5$	$S_c = 6$	$S_c = 2$	$S_c = 1$
3	$S_c = 0$	$S_c = 5$	$S_c = 3$	$S_c = 6$	$S_c = 2$	$S_c = 7$	$S_c = 1$	$S_c = 4$
4	$S_c = 0$	$S_c = 4$	$S_c = 1$	$S_c = 5$	$S_c = 6$	$S_c = 2$	$S_c = 7$	$S_c = 3$
5	$S_c = 0$	$S_c = 2$	$S_c = 5$	$S_c = 7$	$S_c = 1$	$S_c = 3$	$S_c = 4$	$S_c = 6$
6	$S_c = 0$	$S_c = 7$	$S_c = 6$	$S_c = 1$	$S_c = 3$	$S_c = 4$	$S_c = 5$	$S_c = 2$

Table 4.1: Circulation state correspondence table

The second stage encodes the input bit stream once more, but with the encoder starting and ending in the state S_c according to table 4.1.

This double encoding is carried out in the same manner for the sequence in natural order and the interleaved sequence before the W and Y parity bits are added to the final codeword by the puncturing stage. The interleaving and puncturing stages are described in the following subsections.

4.2.3 Interleaving

Interleaving is carried out in two stages, the first uses an intra-couple permutation and the second uses an inter-couple permutation.

In the first stage, for a block of N couples, let

$$j = 0, \dots, N-1 \quad (4.1)$$

If j is even, invert the couple so that (A, B) becomes (B, A) .

In the second stage a permutation parameter P is calculated, P being dependent on the remainder when j is divided by 4, and on P_0 to P_3 which are permutation parameters fixed by the size of N . P_0 to P_3 are listed in table 4.2 and the method of calculating P is shown in table 4.3.

Frame size N , in couples	P_0	P_1	P_2	P_3
48 (12 bytes)	11	24	0	24
64 (16 bytes)	7	34	32	2
212 (53 bytes)	13	106	108	2
220 (55 bytes)	23	112	4	116
228 (57 bytes)	17	116	72	188
424 (106 bytes)	11	6	8	2
432 (108 bytes)	13	0	4	8
440 (110 bytes)	13	10	4	2
752 (188 bytes)	19	376	224	600
848 (212 bytes)	19	2	16	6
856 (214 bytes)	19	428	224	652
864 (216 bytes)	19	2	16	6

Table 4.2: Fixed interleaving permutation parameters

$j \bmod 4$	Calculation of P
0	$P = 0$
1	$P = N/2 + P_1$
2	$P = P_2$
3	$P = N/2 + P_3$

Table 4.3: Calculation of permutation parameter P

Finally an index i can be calculated with N discrete values from 0 to $N-1$, each value mapping to a particular value of j by the following equation:

$$i = (jP_0 + P + 1) \bmod N \quad (4.2)$$

4.2.4 Puncturing

The turbo code defined by the DVB-RCS standard has seven code rates available, each produced by puncturing, i.e. the removal of parity bits. Table 4.4 shows the puncturing pattern for each of the code rates available. A '1' represents the bit being kept and a '0' represents a bit being discarded.

Code Rate	Y puncturing pattern	W puncturing pattern
1/3	1	1
2/5	1 1	1 0
1/2	1	0
2/3	1 0	0 0
3/4	1 0 0	0 0 0
4/5	1 0 0 0	0 0 0 0
6/7	1 0 0 0 0 0	0 0 0 0 0 0

Table 4.4: Puncturing pattern for the DVB-RCS turbo code

Chapter 4 – Duo-Binary Turbo Coding and Decoding

For code rates of $1/2$ and above, the W bits are discarded altogether. So for example, if a puncturing rate of $2/3$ is used, all W bits are discarded and every other Y bit is also discarded.

After puncturing, the codeword is arranged with all the recursive couples A and B first, followed by the Y_1 and Y_2 couples remaining and lastly by any W_1 and W_2 couples that remain as illustrated in figure 4.3.

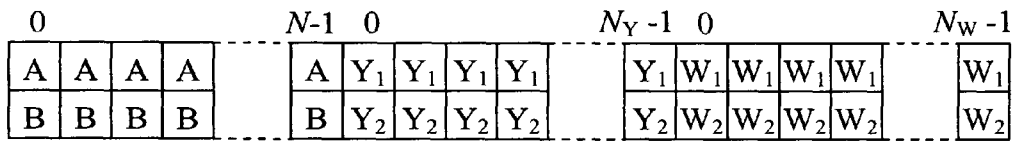


Figure 4.3: Codeword ordering

4.3 Turbo Decoding

4.3.1 Overview

Although the parallel concatenation of turbo codes was a new concept when introduced in [32], the major advantage of turbo codes lies in the decoding process. Where older codes decoded in a single stage, turbo codes use an iterative process combined with an algorithm that yields the *a posteriori* probability (APP) of each of the bits, or couples in the case of a duo-binary code such as the one used here. A suitable algorithm was presented in [108], which requires more processing than the Viterbi algorithm and therefore found little use until the APP was required directly for an iterative process.

Chapter 4 – Duo-Binary Turbo Coding and Decoding

The duo-binary nature of this code yields several significant benefits over binary codes during the decoding process. Firstly, convergence on the corrected codeword happens more quickly with duo-binary codes [35], leading to reduced latency. Secondly, a larger minimum distance is obtainable due to the possibility of intra-symbol permutation which adds an extra degree of freedom in the design of permutations. Thirdly, the trellis has half the number of states as an equivalent binary code of the same constraint length and therefore the constituent decoder requires half the memory and can also be clocked at half the frequency of its binary counterpart. Lastly, when the max-log-MAP approximation is made in order to greatly simplify the decoding process, the amount of coding gain lost is smaller than that of binary codes [109].

4.3.2 Trellis Representation

The trellis is a more meaningful expression of the constituent code for decoding purposes than the block diagram of figure 4.2. The trellis for the DVB-RCS constituent code appears in [110], a modified version of which appears in figure 4.4. Since two bits are encoded at each stage, the trellis has 4 branches leaving each state and the states, the inputs and outputs of the encoder are labelled in octal. Rather than label the encoder outputs on each branch of an already busy diagram, they are labelled to the left of each state and from left to right indicate the top to bottom branches respectively.

So for example, if the encoder is in state 4, representing values of ‘1’, ‘0’ and ‘0’ on the outputs of S_1 , S_2 and S_3 of figure 4.2 and receives an input symbol of 3 (A is ‘1’ and B is ‘1’), the state becomes 5, or in other words, the outputs of S_1 , S_2 and S_3 become ‘1’, ‘0’ and ‘1’ respectively and W and Y are both ‘1’.

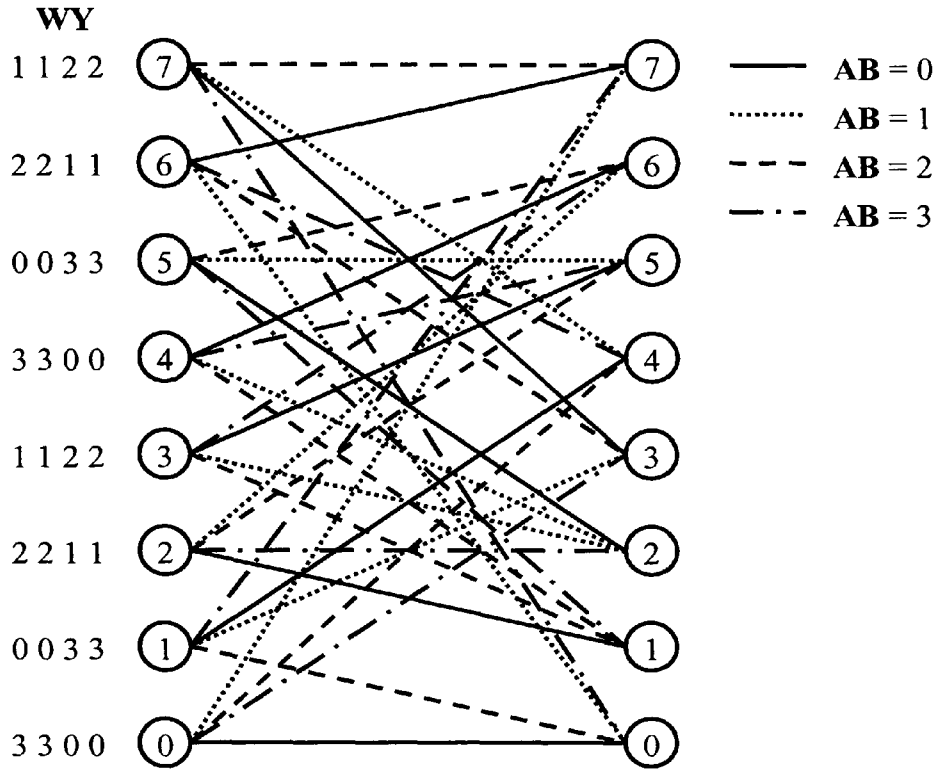


Figure 4.4: Trellis associated with DVB-RCS constituent encoder/decoder

This representation helps to visualise the calculation of the state and branch metrics that is required in the decoding process. Following this representation, the state S_i will be used to denote the i^{th} state, where $i \in \{0, \dots, 7\}$.

4.3.3 Turbo Decoding Process

The DVB-RCS turbo code is decoded by an iterative process, as is the case with all turbo codes. However the situation is complicated slightly for several reasons, principally because the code is duo-binary, but also because the code is circular or tail-biting. A decoding method for this particular code has been outlined in [111] and [110] and is reproduced here. A block diagram of the decoding process is shown in figure 4.5.

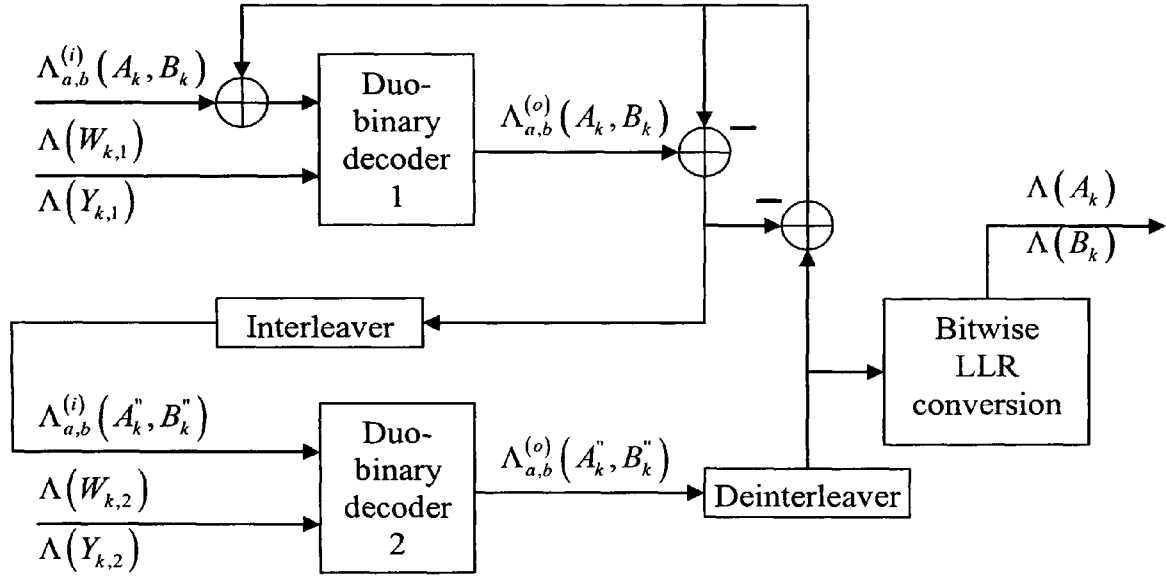


Figure 4.5: A Decoder for the DVB-RCS turbo code

Although the decoding could be performed on the probabilities, the log-probabilities are used instead as this simplifies the hardware implementation of decoders such as that presented in [111] by the use of the max-log-MAP approximation.

Unlike binary codes, the log-likelihood ratios (LLRs) that are presented to the inputs of the duo-binary decoders must contain three values relative to the case where input bits a and b are both zero, one for the case where a is '1' and b is '0', another for the case where a is '0' and b is '1' and a final one for the case where both a and b are '1'. The calculation of the LLRs for couple (A_k, B_k) , where time index $k \in \{0, \dots, N-1\}$, is calculated as follows:

$$\Lambda_{a,b}(A_k, B_k) = \log \left(\frac{P(A_k = a, B_k = b)}{P(A_k = 0, B_k = 0)} \right) \quad (4.3)$$

The function of each of the decoders in the diagram of figure 4.5 is to update the log-likelihoods for each couple in the frame before passing the information to the next stage. As with binary codes, the output from the previous decoding stage is subtracted from the overall LLR value at the input to each decoding stage to generate extrinsic information about the couples. This avoids feeding back information that has already been resolved at a previous stage. In the figure and following discussion, superscript (i) is used to denote the input to the decoder and superscript (o) is used to denote the output of each decoder. Also, superscript quotation marks are used to indicate that the sequence of couples is in interleaved order.

The interleaving and deinterleaving stages must carry out the inter-symbol permutation as described in section 4.2.3 but intra-symbol permutation must be carried out on a symbol-wise basis rather than a bit-wise basis to ensure that the three log-likelihood ratios $\{\Lambda_{0,1}(A_k, B_k), \Lambda_{1,0}(A_k, B_k), \Lambda_{1,1}(A_k, B_k)\}$ associated with each couple are not separated. However, this is a fairly straightforward task as it merely involves swapping the LLRs associated with the $\Lambda_{0,1}(A_k, B_k)$ and $\Lambda_{1,0}(A_k, B_k)$ cases.

The extension of the max-log-MAP algorithm presented in [112] to the duo-binary case involves simply labelling each branch with the LLRs of the systematic and parity couples that are associated with that branch. The max-log-MAP method assumes that $\max^*(x, y) \approx \max(x, y)$ where x and y are LLRs and the \max^* operation, as defined in [113] is:

$$\max^*(x, y) = \max(x, y) + \ln(1 + e^{-|x-y|}) = \ln(e^x + e^y) \quad (4.4)$$

Chapter 4 – Duo-Binary Turbo Coding and Decoding

The max operation simply takes the maximum of the two arguments. If extended to more than two LLRs, the relationship becomes a recursive operation of max* pairs:

$$\max^*(x, y, z) = \max^*[\max^*(x, y), z] = \ln(e^x + e^y + e^z) \quad (4.5)$$

Although the max* operator gives better performance than the max approximation, the loss in gain for duo-binary codes is only between 0.2 and 0.3 dB [110]. Hardware implementation is vastly simplified when the max approximation is used as it allows decisions using only a comparator, where the max* operator would require additional processing. The max operation is used in the following explanation of the forward and backward recursion for the state metric calculation, although it should be noted that it can be substituted for the max* operator.

As with the APP method as applied to binary turbo codes, decoding requires the calculation of forward and backward recursions of the state metrics (i.e. the log-likelihoods of any state being the state of the encoder at time k) for the entire frame. The calculation of both recursions first requires knowledge of the branch metrics (i.e. the log-likelihoods of any branch being that taken by the encoder at time k). This depends on the message and parity couples associated with each branch (see AB and WY for the trellis in figure 4.4), as well as the extrinsic information passed to the decoder. If we let $\gamma_k(\mathbf{S}_i \rightarrow \mathbf{S}_j)$ denote the branch metric at time k corresponding to the state transition of $\mathbf{S}_i \rightarrow \mathbf{S}_j$, and a , b , w and y equal the binary equivalents of the octal representation of AB and WY associated with the trellis, then

$$\gamma_k(\mathbf{S}_i \rightarrow \mathbf{S}_j) = \Lambda_{a,b}^{(i)}(A_k, B_k) + w\Lambda(W_k) + y\Lambda(Y_k) \quad (4.6)$$

Let us now define the forward recursion (\mathbf{S}_0 to \mathbf{S}_{N-1}) as α and the backward recursion (\mathbf{S}_{N-1} to \mathbf{S}_0) as β . If $\alpha_k(\mathbf{S}_i)$ is the normalised forward metric at time k and for state number i and $\alpha'_{k+1}(\mathbf{S}_j)$ is the forward metric to be calculated at time $k+1$ and in state number j prior to normalisation, then:

$$\alpha'_{k+1}(\mathbf{S}_j) = \max \left\{ \alpha_k(\mathbf{S}_i), \gamma(\mathbf{S}_i \rightarrow \mathbf{S}_j) \right\} \quad (4.7)$$

where the max operation is carried out over the four state transitions leading to state \mathbf{S}_j at time $k+1$. The forward metrics are then normalised with respect to the metric of \mathbf{S}_0 :

$$\alpha_{k+1}(\mathbf{S}_j) = \alpha'_{k+1}(\mathbf{S}_j) - \alpha'_{k+1}(\mathbf{S}_0) \quad (4.8)$$

If $\beta_{k+1}(\mathbf{S}_j)$ is the normalised backwards metric with respect to state zero for state number j at time $k+1$ and $\beta'_k(\mathbf{S}_i)$ is the backwards state metric to be calculated at time k and in state number i prior to being normalised, then:

$$\beta'_k(\mathbf{S}_i) = \max \left\{ \beta_{k+1}(\mathbf{S}_j), \gamma(\mathbf{S}_i \rightarrow \mathbf{S}_j) \right\} \quad (4.9)$$

where the max operation is carried out over the four state transitions exiting \mathbf{S}_i at time k . The backwards metrics are then normalised to the metric of \mathbf{S}_0 in the same manner as the forward metrics.

$$\beta_k(S_i) = \beta'_k(S_i) - \beta'_k(S_0) \quad (4.10)$$

If the code were terminated in the usual way, this methodology would suffice with enough extra trellis sections to account for the terminating sequence of zeros. However, as mentioned earlier the tail-biting nature of the code changes the decoding process slightly and in fact, termination of the trellis of figure 4.4 is not even possible by applying extra zeros at the end of a bit sequence.

Although no information is known about the starting state due to trellis termination, it is certain from the dual encoding process that the final state of the encoder is equal to that of the starting state. For this reason, the trellis can be seen as cylindrical rather than flat, the final state wrapping back and becoming the initial state. A diagram of this to aid with visualisation appears in figure 4.6, using the forward metric α as an example.

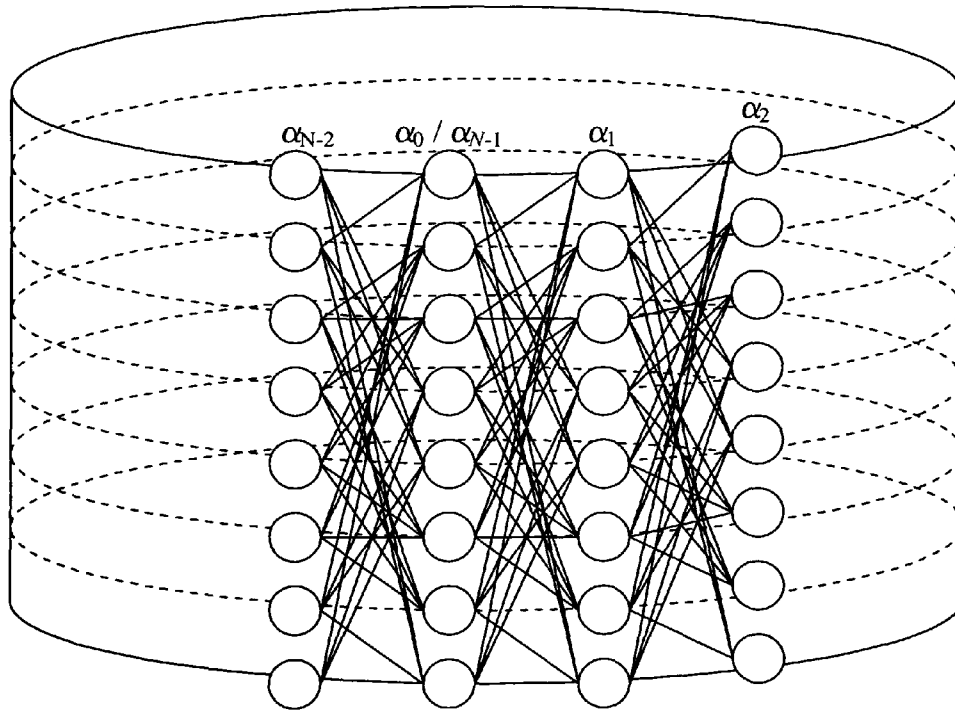


Figure 4.6: Cylindrical trellis for tail-biting DVB-RCS turbo code

Chapter 4 – Duo-Binary Turbo Coding and Decoding

Three methods of decoding tail-biting trellises are presented in [114], the last of which has been used to obtain the α and β metrics in this case.

α_0 is first initialised so that each state has an equal log-likelihood. If there are M possible states then:

$$\alpha_0 = \log\left(\frac{1}{M}, \frac{1}{M}, \dots, \frac{1}{M}\right) \quad (4.11)$$

The forward metrics are calculated as normal starting from α_0 , where the state metric is equally likely for each state. Once a full cycle around the cylinder has been completed, the forward recursion continues around the cylinder a second time, replacing the previous values of α with the new ones after comparing them. When $|\alpha_{k+N-1} - \alpha_k|$ is sufficiently small for all states, the process stops.

A similar procedure is then executed in the reverse direction. β_{N-1} is first initialised so that all M states are equally likely:

$$\beta_{N-1} = \log\left(\frac{1}{M}, \frac{1}{M}, \dots, \frac{1}{M}\right) \quad (4.12)$$

The reverse metrics are then calculated as normal starting from β_{N-1} and working backwards. Once a full cycle has been completed, the backwards recursion continues to calculate new values of β , replacing the previous values until $|\beta_k - \beta_{k+N-1}|$ is sufficiently small for all states at which point the recursion stops.

A stopping criterion of 0.1 dB was chosen for the magnitude of the difference between the previous metrics and the most recent metrics. Generally, the process was found to converge very quickly, where the wrap depth W was just a small fraction of N , i.e. just a few sections of the trellis. However, the stopping criteria was incremented by a further 0.1 dB each time a full cycle was completed in order to avoid the scenario of endless looping where the stopping criteria was never met.

With the forward and backward metrics stored the LLRs can be calculated. Firstly, the likelihood of each branch is calculated by combining the forward metric, the reverse metric and the branch metric of the appropriate section as follows:

$$Z_k(\mathbf{S}_i \rightarrow \mathbf{S}_j) = \alpha_k(\mathbf{S}_i) + \gamma_k(\mathbf{S}_i \rightarrow \mathbf{S}_j) + \beta_{k+1}(\mathbf{S}_j) \quad (4.13)$$

Following this, the likelihood that couple $(A_k, B_k) = (a, b)$ is calculated using the max operator for each branch labelled with (a, b) (see the octal representation for the trellis of figure 4.4).

$$\lambda_k(a, b) = \max_{\mathbf{S}_i \rightarrow \mathbf{S}_j: (a, b)} \{Z_k\} \quad (4.14)$$

Finally, the LLRs at the encoders output can be found.

$$\Lambda_{a,b}^{(o)}(A_k, B_k) = \lambda_k(a, b) - \lambda_k(0, 0) \quad (4.15)$$

where $(a, b) \in \{(0, 1), (1, 0), (1, 1)\}$.

This describes the action of a single decoder block from the block diagram of figure 4.5. The process is identical for both decoders, albeit with different W and Y parity bits and with the extrinsic information in a different order. After a fixed number of iterations the decoding process must stop and the couple-wise LLRs relative to the case where a and b are both zero must be converted into bitwise LLRs. For simulation purposes, the stopping criterion can also be when the correct bit sequence is decoded, as this saves processing time, although this information would obviously not be available in a real decoder. This process again makes use of the max operator:

$$\Lambda(A_k) = \max \left\{ \Lambda_{1,0}^{(o)}(A_k, B_k), \Lambda_{1,1}^{(o)}(A_k, B_k) \right\} - \max \left\{ \Lambda_{0,0}^{(o)}(A_k, B_k), \Lambda_{0,1}^{(o)}(A_k, B_k) \right\} \quad (4.16)$$

$$\Lambda(B_k) = \max \left\{ \Lambda_{0,1}^{(o)}(A_k, B_k), \Lambda_{1,1}^{(o)}(A_k, B_k) \right\} - \max \left\{ \Lambda_{0,0}^{(o)}(A_k, B_k), \Lambda_{1,0}^{(o)}(A_k, B_k) \right\} \quad (4.17)$$

Once the bit-wise LLRs have been found, obtaining the bit sequence is a simple matter of mapping any positive LLRs as ‘1’s and any negative LLRs as ‘0’s.

4.4 Software Implementation

A turbo encoder and decoder for the DVB-RCS code were implemented in Matlab and tested to determine the performance of the code in terms of code rate and block size. The code was also compared against a standard convolutional code to determine the improvement achieved.

The turbo encoder was a simple case of defining the trellis of figure 4.4 within Matlab and calling a standard Matlab function from the communication toolbox that

convolutionally encodes a sequence of bits by the trellis definition. However, no function is provided within Matlab to decode turbo codes and this software had to be coded in full. The general decoder of figure 4.5 was written as an m-file in Matlab's native format. However, the nature of this type of code is not well suited to interpreted languages as the multiple nested loops of the decoder blocks and interleaving and de-interleaving blocks of figure 4.5 execute very slowly. To improve the execution speed, these were written as C mex files that could be compiled into a dynamic link library and called from the Matlab environment in the same manner as a normal m-file function. A software flow diagram of the process shown in figure 4.5 is shown in figure 4.7, where the input is the bit-wise LLRs received for the block.

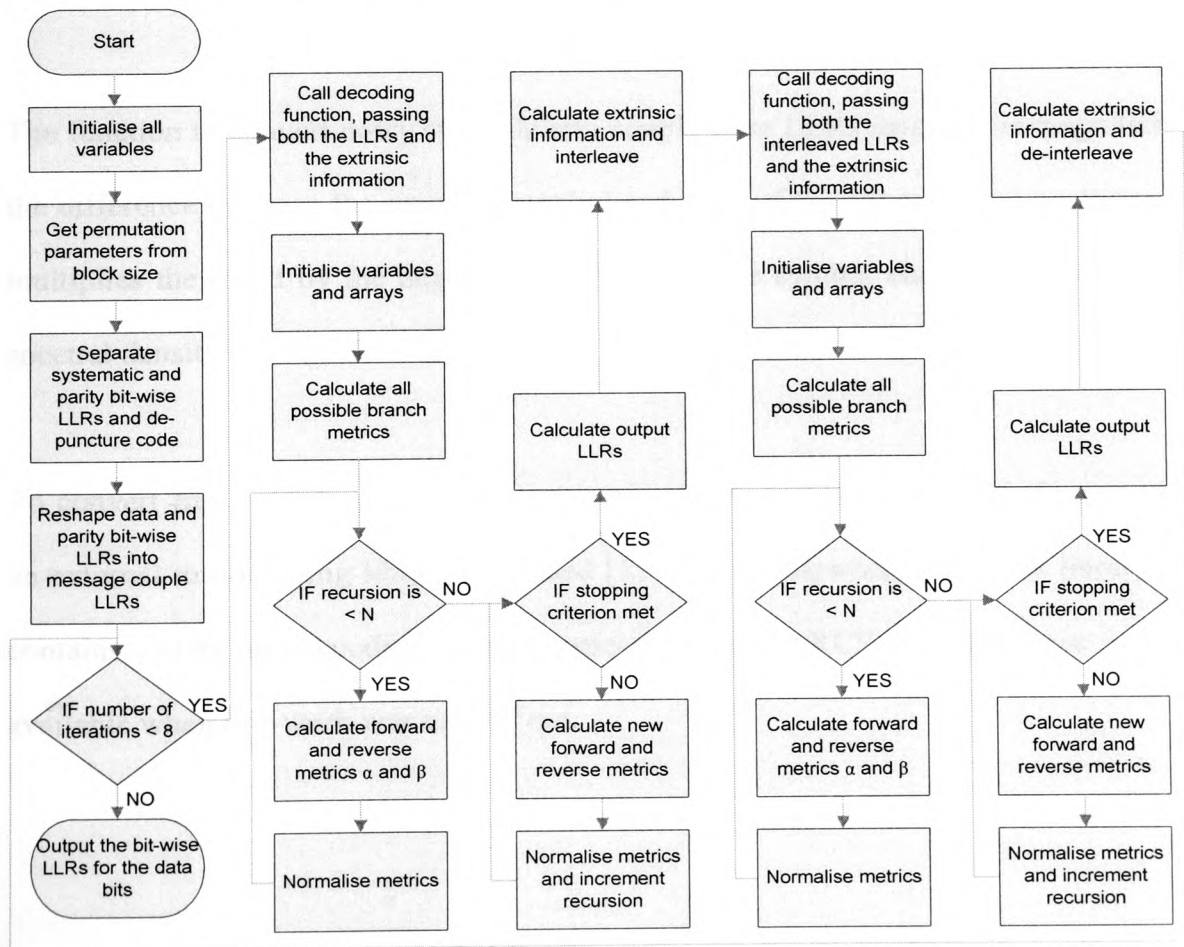


Figure 4.7: Software flow diagram of turbo-decoding process

Chapter 4 – Duo-Binary Turbo Coding and Decoding

The Matlab m-file used for the decoding process appears in appendix B. The Matlab mex files for the decoder block, the interleaver block, the de-interleaver block and the permutation look-up appear in appendices C, D, E and F respectively.

In addition to the basic elements of the decoder, several other functions were needed to perform conversions between bit-wise and couple-wise LLRs and to convert noisy symbols in signal space into LLRs. The function to convert bit-wise to couple-wise LLRs simply adds the individual bit-wise elements corresponding to the appropriate ‘1’ position, i.e. $\Lambda_{0,0}(A_k, B_k) = 0$, $\Lambda_{0,1}(A_k, B_k) = \Lambda(B_k)$, $\Lambda_{1,0}(A_k, B_k) = \Lambda(A_k)$, $\Lambda_{1,1}(A_k, B_k) = \Lambda(A_k) + \Lambda(B_k)$. This code appears in appendix G.

The function to convert noisy symbols into couple-wise LLRs squares the magnitude of the difference between the received symbol and each of the constellation positions and multiplies the result by the negative equivalent of the symbol energy per noise power spectral density (E_s/N_0). This code appears in appendix H.

To convert from couple-wise LLRs to bit-wise LLRs, a function called ‘Somap’ from an external turbo-coding library was used [115]. It is noteworthy that this library now contains coding and decoding elements specific to DVB-RCS although these were not available when this work was carried out.

4.5 Performance

The performance of the DVB-RCS turbo code and the decoding process described above was investigated in order to set bit energy per noise power spectral density (E_b/N_0) thresholds for future simulations that use the code, or simply assume its use.

Firstly, a constant block size of 212 couples, or 424 bits was encoded with each of the available coding rates to determine the E_b/N_0 threshold for a BER of 10^{-6} for each of the rates. This block size was chosen as it corresponds to a 53-byte ATM cell. Bit error rate and frame error rate (FER) statistics were obtained for each code rate, the results of which appear in figures 4.8 and 4.9 respectively. Only the upper portion of E_b/N_0 values were simulated in order to process more frames, since the lower values of BER are more useful for designing communications links.

From figure 4.8 it is obvious that if a nominal code rate of 6/7 is used for a link then an E_b/N_0 of approximately 5.5 dB is required to communicate with a BER of 10^{-6} . It is also clear that the effect of decreasing code rate increases gain by almost the same margin at all values of BER. The FER values of figure 4.9 were checked against the results of others who have simulated the performance of this particular turbo code [110, 116], and some 0.3 dB difference in coding gain was found in the worst case, the code here underperforming in comparison with the other results. Very little information regarding all the parameters used was available from either of the resources and the source of the discrepancy could not be determined despite many permutations of adjustments to all the elements involved in the decoding process. Finally it was decided to use these more pessimistic values in the final stages of the design.

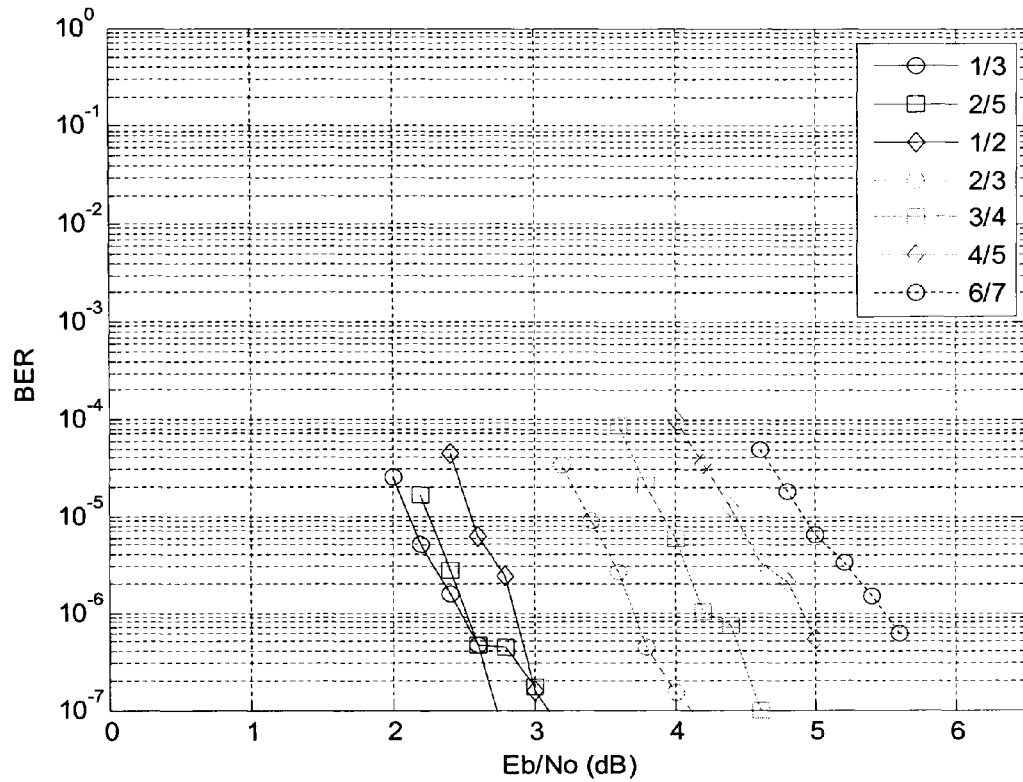


Figure 4.8: Bit error rates for DVB-RCS turbo code for all available code rates with N=212 and 8 decoding iterations

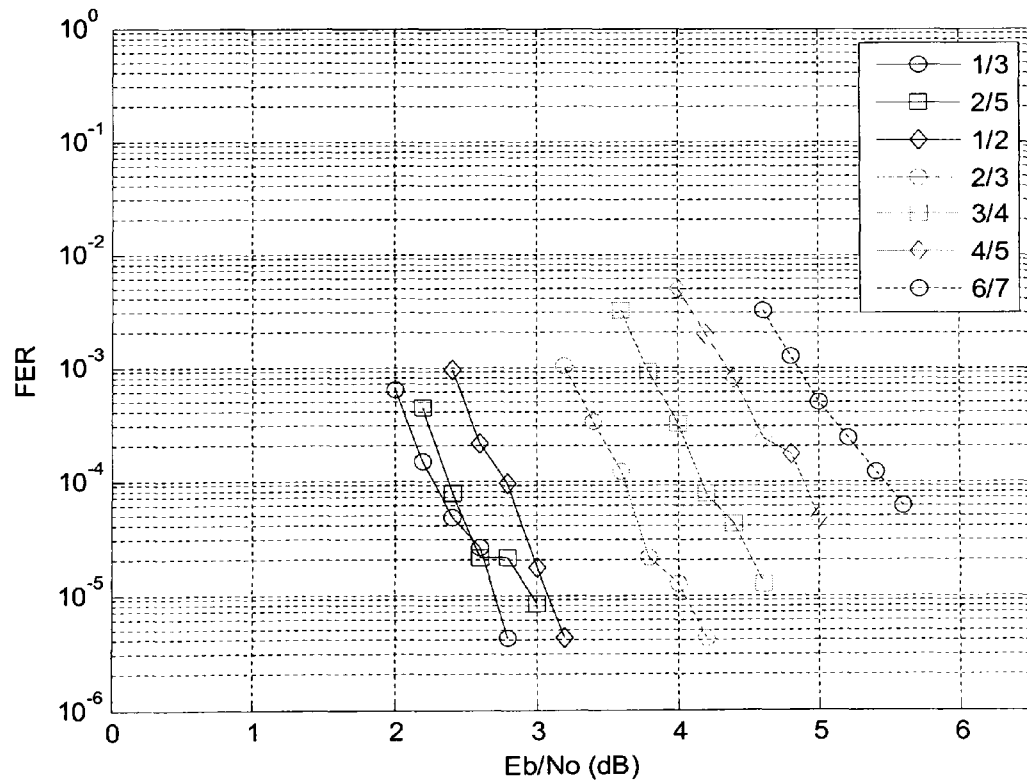


Figure 4.9: Frame error rates for DVB-RCS turbo code for all available code rates with N=212 and 8 decoding iterations

The second set of results used a constant code rate of $\frac{1}{2}$ but varied the block size. Block sizes of 48, 212, 424 and 848 were simulated, the results of which appear in figures 4.10 and 4.11. Unlike the coding gain achieved by decreasing the code rate, increasing block size gives varying performance gains depending on the BER of interest. At a BER of 10^{-6} , the case where $N=48$ will have approximately 2 dB less coding gain than that of an ATM cell, whereas a cell double the size will have approximately 0.2 dB of gain over that of an ATM block. Since the steepness of the curve remains unchanged for changes in coding rate, it is assumed that all the various permutations of coding rate and block size need not be simulated, and these coding gain assumptions apply for all coding rates.

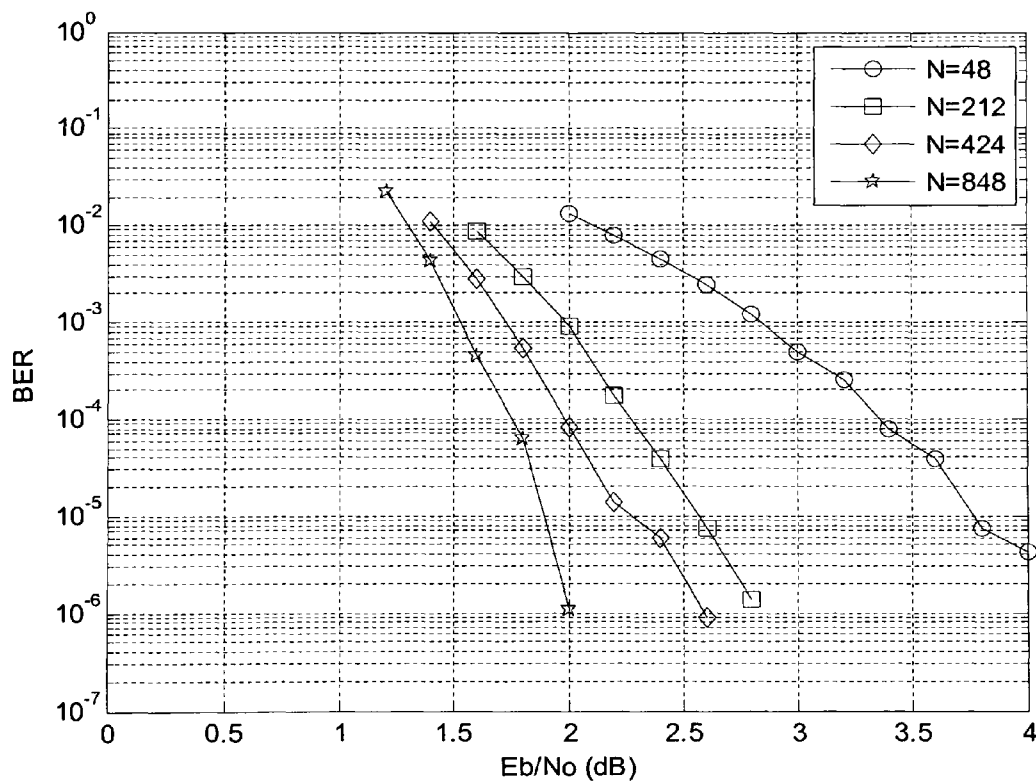


Figure 4.10: Bit error rates for various block sizes at code rate $\frac{1}{2}$ and with 8 decoding iterations

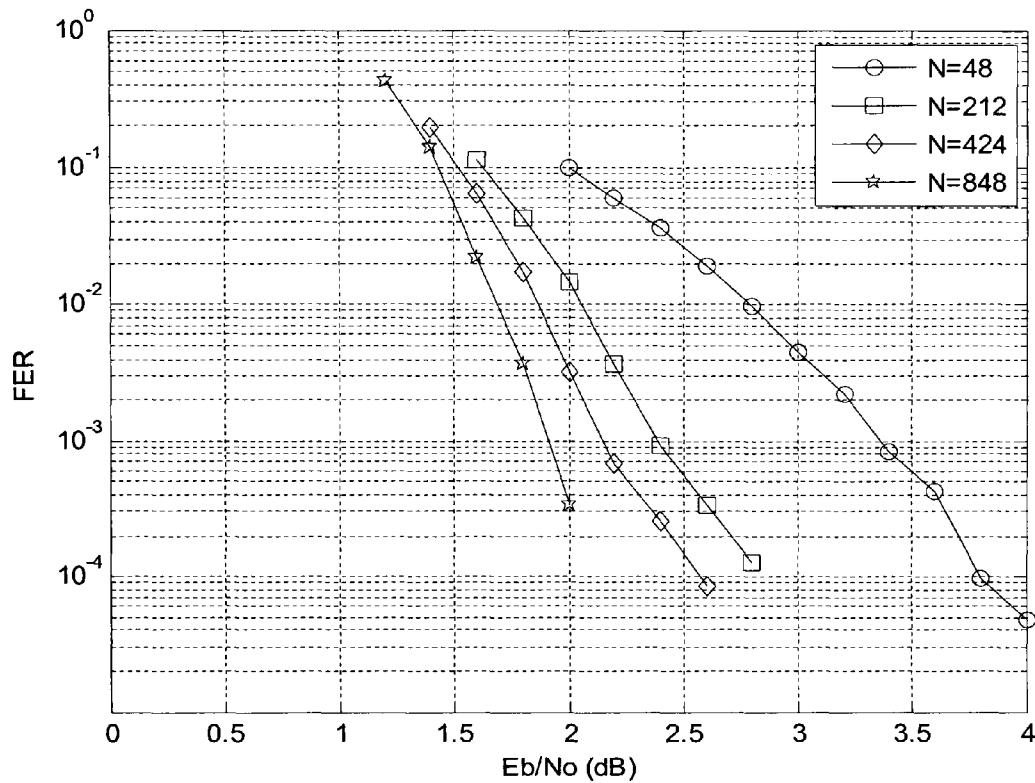


Figure 4.11: Frame error rates for various block sizes at code rate $\frac{1}{2}$ and with 8 decoding iterations

Finally, a $\frac{1}{2}$ rate DVB-RCS turbo code was compared with a $\frac{1}{2}$ rate convolutional code with constraint length 7 and soft Viterbi decoding for a block size of 212 couples. The results appear in figures 4.12 and 4.13.

At a BER of 10^{-6} , a coding gain improvement of over 5 dB is possible for ATM cell size blocks, assuming random errors. This is a very significant improvement over previous codes and one of the enabling factors for Ka and V band commercial satellite applications.

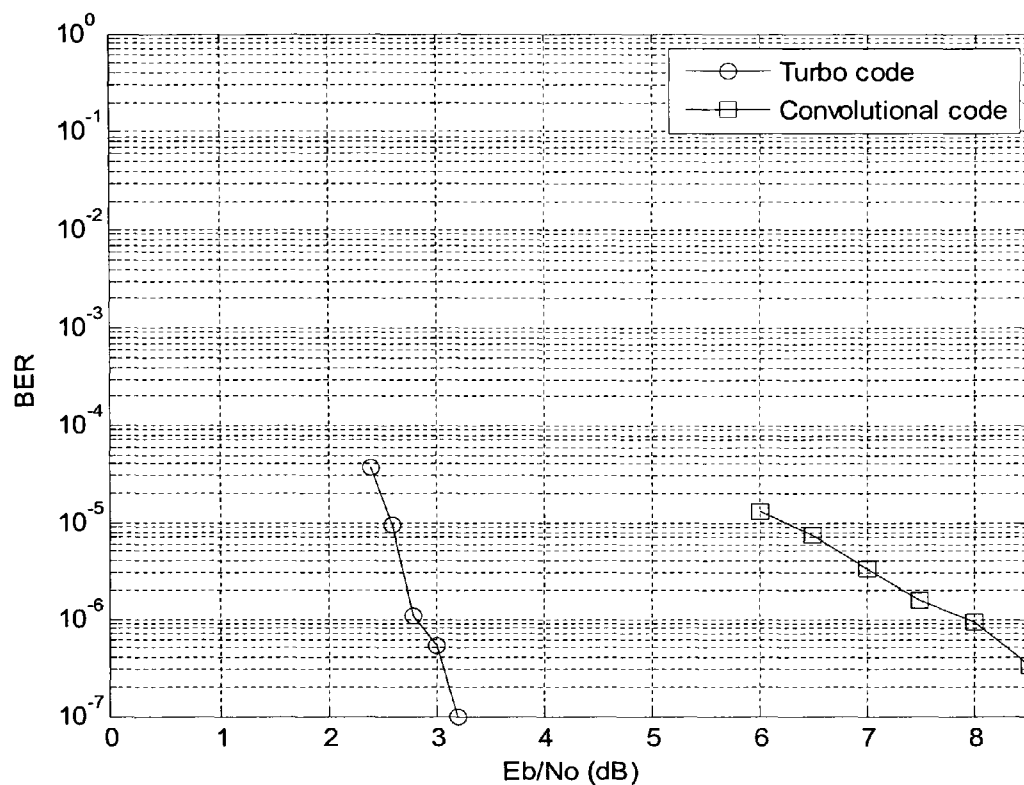


Figure 4.12: Bit error rate comparison for $\frac{1}{2}$ rate DVB-RCS turbo code and $\frac{1}{2}$ rate convolutional code with constraint length 7 and soft Viterbi decoding

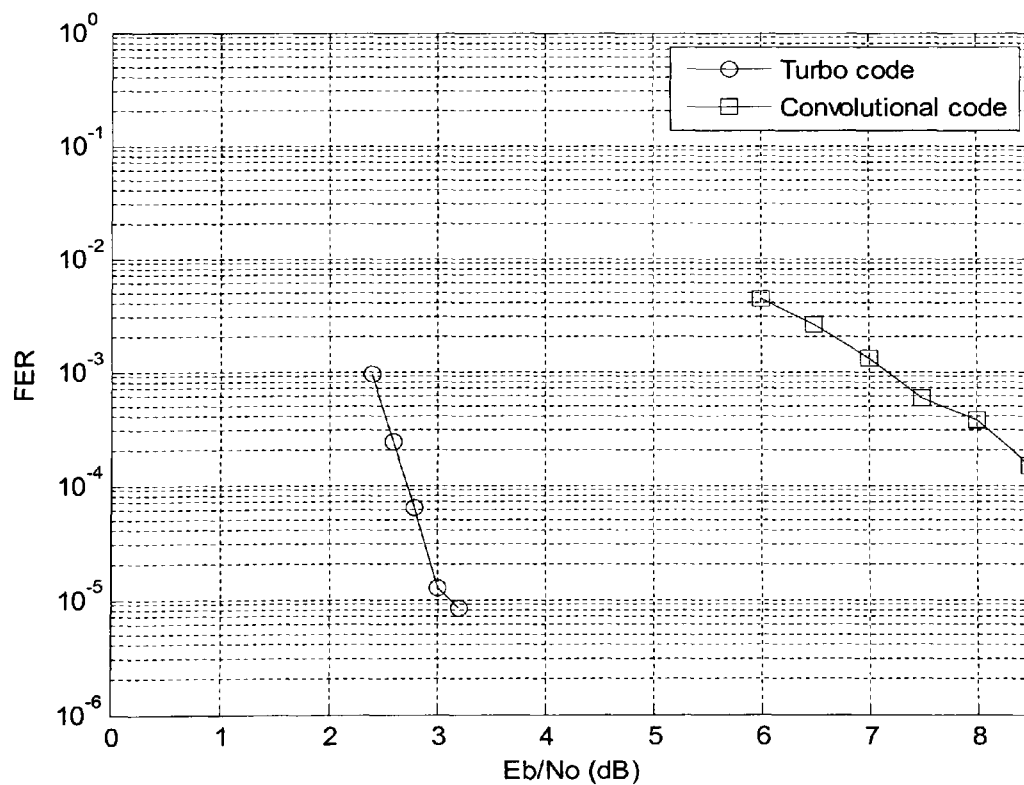


Figure 4.13: Frame error rate comparison for $\frac{1}{2}$ rate DVB-RCS turbo code and $\frac{1}{2}$ rate convolutional code with constraint length 7 and soft Viterbi decoding

4.6 Conclusions

The traffic in the final simulations was sent via ATM packets of 53 bytes, or 212 couples at code rates between $1/2$ and $6/7$. Also control information was sent on the forward channel with a $1/2$ rate code in packets of 424 and 48 couples, while the control information on the return channel utilized a standard $1/2$ rate convolutional code.

For the cases in the final simulation where the coding was not implemented in full, the E_b/N_0 requirement was instead used to assess the availability of links. This was read from the graph in figure 4.8 for the traffic information and from the graph in figure 4.10 for the control information. This remains a reliable method for link budget design without having to simulate the coding and decoding of each ATM packet since on average the bit error rates on the link will be the same as those found here.

A more detailed discussion of the channel design can be found in chapter 7.

Chapter 5

Site Diversity Approximation

5.1 Introduction

The previous chapter described the implementation of an adaptive coding scheme for use with the proposed system of satellite internet delivery. However, another of the chosen FMTs whose implementation is non-trivial is site diversity. Ideally, two time series of attenuation measured at two diversity stations would be used in determining the attenuation on the link at any point in time. However, since these measurements have not been carried out for the data that is available, another albeit less accurate method of estimating the instantaneous gain due to site diversity had to be implemented.

This left two options for implementing site diversity, firstly to use a measured time series combined with a simulated time series, or secondly to use a single time series with some method of SNR improvement due to site diversity. Of these two options the first resulted in two problematic issues; the overall accuracy of simulated time series in comparison to real data and also the lack of any method to correlate the simulated time series with the measured time series given attenuation time series simulators such as [117, 118]. This would have led to results which were less accurate due to the nature of simulated time series but also the assumption of very large separation distance would have to be made due to the decorrelation of the two time series. The second option on the other hand would use the measured time series but provide some kind of correlated value of attenuation for the diversity station.

The following section describes the method used to derive a second attenuation value suitable for a diversity station along with the assumptions made by this process. Results of this method are presented in the third and final section.

5.2 Dual-site Concurrent Rain Rate Analysis

While no measurements of attenuation for dual-site diversity were available, a hydrological RADAR experiment (HYREX) [14] was carried out several kilometres away from the site where the attenuation time series from the ITALSAT satellite was gathered.

The methodology herein details the use of a single time series of attenuation to generate a statistically sound dataset for a diversity system simulation. This is achieved by shifting the diversity site in time along the time series so that the time shift approximates a physical distance separating both sites. Theoretically this limits the system to a single pattern of weather, i.e. moving directly from one site to the other, without any change. However, since the sites are not separated by a distance large enough to alter the prevailing climatic conditions, the ratio of stratiform to convective rain events at both sites should be approximately equal. The randomness of rain types and movement at both sites should leave a simulated system with a fixed rain pattern having statistics that are close to real diversity statistics, provided the time series represents an average year. This is advantageous over simply including a diversity gain for the time series, since the amount of gain varies with availability (see figure 2.4).

The HYREX measurement campaign was carried out over a 135 km² area in Southern England where the rain rates from 52 tipping bucket rain gauges were recorded over a three year period. For the purpose of a proposed diversity system, two gauges in a near-optimum configuration were selected, which were the Bridge Farm Orchard (BFRM) gauge and the Goodedge Farm Fields (GOOD) gauge. This gave a proposed diversity

Chapter 5 – Site Diversity Approximation

system of separation 9.6 km and an 82° baseline with respect to the ITALSAT satellite, from which the Sparsholt attenuation time series were obtained. Each gauge measured the rain rate in ten second intervals, but these time series were averaged over five minutes to bring the rain rate to an acceptable minimum resolution of 2.4 mm/h as the minimum resolution in ten second intervals was 72 mm/h. Five minute samples were suggested in [119] as being optimum for reducing rain rate resolution without adversely affecting the rain rate values, i.e. without averaging too heavily which would reduce high rain rates disproportionately.

Joint statistics of rain rates at both sites were gathered from the time series. Table 5.1 shows joint statistics of this analysis, the rows indicating the rain rate at the BFRM site and the columns indicating the rain rate at the GOOD site, with the number in each cell corresponding to the number of samples. Rain is then classified into types to make the statistics more manageable. No rain is defined as anything under 2.4 mm/h, light rain is anything between 2.4 mm/h and 4.8 mm/h and heavy rain is anything above 4.8 mm/h.

In the colour scheme for table 5.1, a black background with white text indicates no rain at either site from here on denoted (0,0), in which, as expected, the majority of samples fall. An almost black background with white text indicates light rain at one site, no rain at the other, from here on denoted (0,1) (this encompasses both conditions of light rain at the GOOD with no rain at the BFRM site and no rain at the GOOD site and light rain at the BFRM site). A dark grey background with white text indicates light rain at both sites, from here on denoted (1,1). A dark grey background with black text indicates heavy rain at one site and no rain at the other, from here on denoted (0,2) (again both permutations are included). A light grey background with black text indicates heavy

Chapter 5 – Site Diversity Approximation

rain at one site and light rain at the other, from here on denoted (1,2) (both permutations are included). Finally, a white background with black text indicates heavy rain at both sites, from here on denoted (2,2).

The purpose of this analysis is to quantify the ratio of concurrent heavy rain events to the total number of heavy rain events, and the ratio of concurrent light rain events to the total number of light rain events in order to guide the subsequent analysis of the attenuation time series from Sparsholt.

The ratio of concurrent heavy rain to total heavy rain is calculated as the ratio of the sum of (2,2) with the sum of (2,2), (1,2) and (0,2) (or 150 divided by the sum of 150, 323 and 617, if the elements are summed from table 5.1). This results in a value of 0.1376 (150 divided by 1090). The ratio of concurrent light rain to total light rain is calculated as the ratio of the sum of (1,1) and (1,2) with the sum of (1,1), (1,2) and (0,1). This results in a value of 0.2312 (963 divided by 4166). These results make sense in light of the fact that convective rain has a small cell extent in comparison to stratiform rain, and therefore the separation distance of 9.6 km is more likely to statistically decorrelate convective events than stratiform events, hence the lower ratio of concurrent heavy rain than concurrent light rain.

mm/h	0	2.4	4.8	7.2	9.6	12	14.4	16.8	19.2	21.6	24	26.4	28.8	31.2	33.6	36	38.4	40.8	43.2	45.6	48	50.4	52.8	55.2	57.6	60
0	100	187	1690	210	70	23	18	6	4	5	2	0	2	3	4	1	0	1	1	0	1	0	1	0	0	0
2.4	1513	640		131	30	14	8	5	2	2	1	0	2	1	0	1	0	0	0	0	0	0	0	0	0	0
4.8	153	86	42	22	9	3	4	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
7.2	55	13	11	10	6	4	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9.6	19	7	5	2	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	12	6	4	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14.4	5	4	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16.8	5	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
19.2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21.6	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	2	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26.4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28.8	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31.2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33.6	3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40.8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43.2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45.6	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50.4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
52.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
55.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
57.6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Colour Key

No rain at either site
Light rain at one site
Concurrent light rain
Heavy rain at one site
Concurrent light and heavy rain
Concurrent heavy rain

Table 5.1: Joint statistics of rain rate at the BFRM gauge (rows) and the GOOD gauge (columns)

Chapter 5 – Site Diversity Approximation

The second stage of the process involved analysis of the attenuation time series from Sparsholt. The threshold rain rates were converted into threshold attenuations for the proposed link to ITALSAT using the ITU-R model for specific attenuation [15] and assuming that rain fills the entire path length between the antenna and the 0° isotherm. This assumption can potentially over-estimate the attenuation on a given link since rain is not always present over the entire path. Even so, it suffices as a first approximation in the absence of co-located radiometers. The height above sea level is taken as 0.039 km, which corresponds to the actual height above sea level of the BFRM site. This finally yields a rainy path length of 6.642 km and attenuation thresholds for heavy rain of 9.5 and 13.2 dB at 40 and 50 GHz respectively and attenuation thresholds for light rain of 5.0 and 7.2 dB at 40 and 50 GHz respectively.

Each attenuation time series was averaged over 30 second samples to eliminate scintillation effects. A matrix was created to store the 6 different fading scenarios up to 30 minutes ahead, resulting in 60 samples for each scenario. Each attenuation time series was opened in chronological order and starting at the first sample the fading condition at the first sample was compared with each of the following 60 samples, the bin in the matrix for the appropriate fading condition being incremented accordingly. The columns 1 to 6 represented the conditions (0,0), (0,1), (1,1), (0,2), (1,2) and (2,2) respectively. For example, if at the reference sample at 40 GHz there was no rain (less than 5 dB), and at the next sample there was light rain (between 5 and 9.5 dB) the bin in row 1 and column 2 of the results matrix, corresponding to one sample ahead and the (0,1) condition, would have been incremented. Then, if at the second sample ahead there was heavy rain (above 9.5 dB), the bin in row 2, column 5 would have been incremented and so on up until 60 samples ahead of the reference. After this had been

Chapter 5 – Site Diversity Approximation

completed the reference sample was incremented to the next in the series and the process started over.

This was carried out in full for the years of 1st August 1999 to 31st July 2000 and 1st September 1997 to 31st August 1998. The results were then used to generate the same statistics for each 30 second sample ahead as had been obtained from the HYREX rain gauges, i.e. the ratio of concurrent heavy rain to all heavy rain and the ratio of concurrent light rain to all light rain were calculated for all 60 separation times (30 seconds to 30 minutes, see above for how the statistics are obtained from the fading scenarios). The ratios were higher at shorter separation times and became lower as more time elapsed, as would be expected since the further away in time from the reference, the less well correlated the rain would become. Tables 5.2 and 5.3 show the statistics for the 1999-2000 year, centred on the optimum correlated heavy rain ratio for the 40 and 50 GHz time series respectively.

Time (mins)	Light ratio	Heavy ratio
14	0.23077849	0.14668133
14.5	0.22888382	0.14124088
15	0.22574067	0.1358518
15.5	0.2237536	0.13051338
16	0.22030858	0.12725306

Table 5.2: Ratio of concurrent light rain to total light rain and concurrent heavy rain to total heavy rain for the 40 GHz attenuation time series

Time (mins)	Light ratio	Heavy ratio
13.5	0.2167649	0.14439946
14	0.21324971	0.14016807
14.5	0.20981211	0.13672922
15	0.2057061	0.13255426
15.5	0.20228958	0.12840985

Table 5.3: Ratio of concurrent light rain to total light rain and concurrent heavy rain to total heavy rain for the 50 GHz attenuation time series

From both tables it is clear that for the 40 and 50 GHz attenuation time series the ratios are very similar. Taking the heavy rain concurrency ratio closest to that of the HYREX

Chapter 5 – Site Diversity Approximation

scenario, which is the ratio that is of most interest of the two, a time separation of 15 and 14.5 minutes was found to be optimum for the 40 and 50 GHz time series respectively with heavy rain concurrency of 0.1358 and 0.1367 for the 40 and 50 GHz time series respectively, compared with 0.1367 for the HYEX scenario. The light rain concurrency is also very close to the HYREX measurements at these time separations, with values of 0.2257 and 0.2098 for the 40 and 50 GHz time series respectively compared with 0.2312 for the HYREX scenario. A final time separation of 14.5 minutes was chosen since the graph for the 40 GHz time series at 14.5 minutes fitted the ITU-R model more closely than that of the 50 GHz time series at 15 minutes.

A 14.5 minute separation over 9.6 km indicates an average storm velocity of 11 m/s or 39.7 km/h, which corresponds well with the average storm velocity found in [120].

5.3 Results

To test the accuracy of a diversity system simulation using these time-shifted versions of the same attenuation time series, a Matlab program was written to record the single site yearly rain attenuation CDF and the simulated joint site CDF and to compare the results with the ITU-R model for site diversity [9]. The program was also written such that the end of one day would wrap into the next day, taking samples from the next time series. Although the days are not all contiguous, it was useful as a test of how well the time series represented a contiguous one as this was the intended course for further simulations. The results are shown in figures 5.1 and 5.2 for the 40 and 50 GHz time series respectively, and are compared with the ITU-R model predictions.

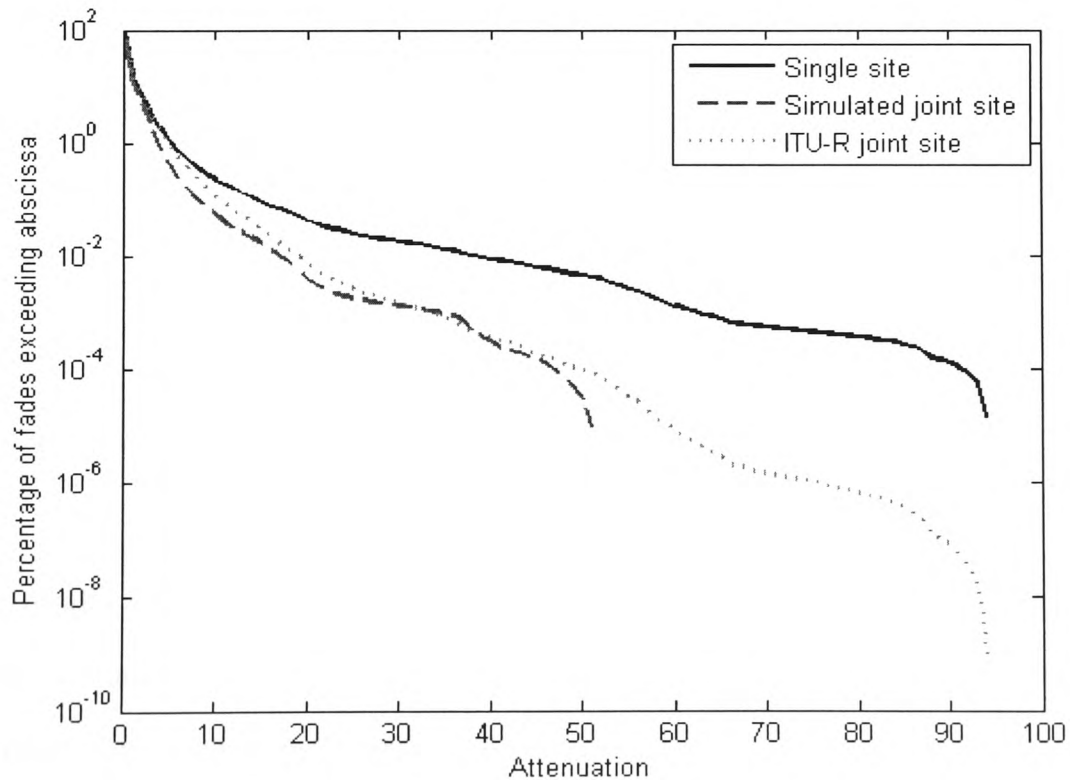


Figure 5.1: Rain attenuation CDF at 40 GHz

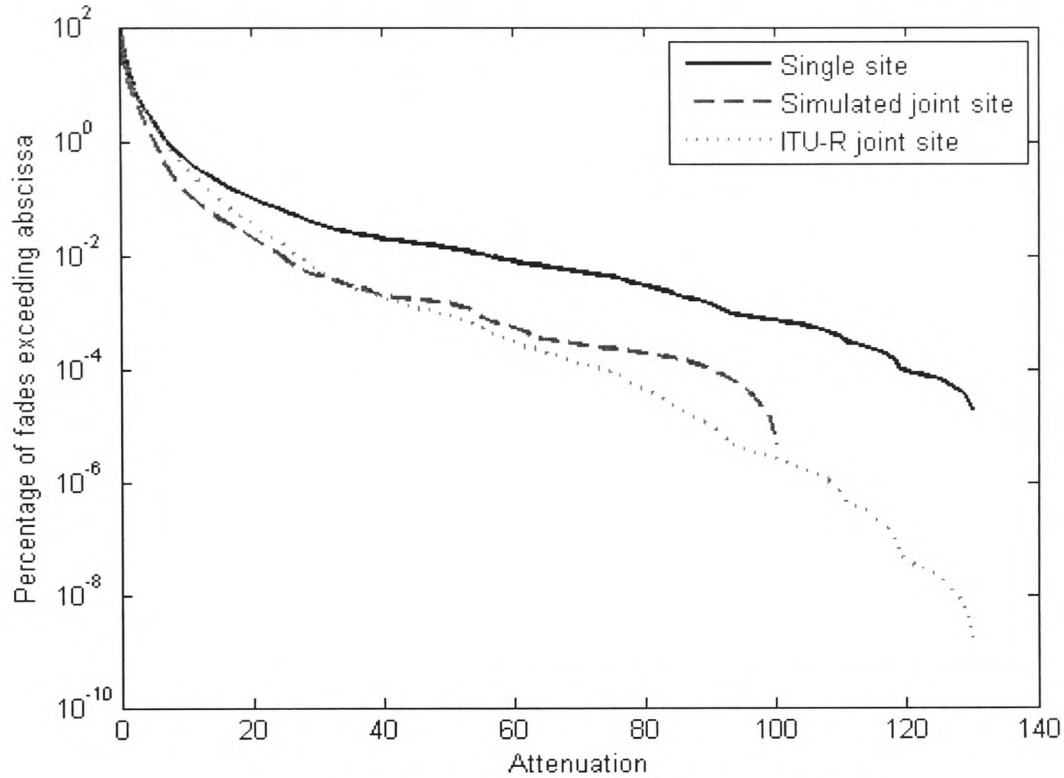


Figure 5.2: Rain attenuation CDF at 50 GHz

The results at both frequencies reveal a satisfactory fit with the ITU-R model for site diversity, giving a time-varying gain that results in a CDF that approaches what would be expected, given the model prediction.

5.4 Conclusions

The results of this analysis were utilised throughout the simulations that appear in chapters 8 and 9 to approximate the effect of site diversity for two hub stations in the configuration stated at the beginning of this chapter. Given a primary hub station with a measured time series, $A(t)$, the attenuation experienced at the secondary hub was $A(t +$

Chapter 5 – Site Diversity Approximation

t_0), where t was 870 seconds (14.5 minutes). A final hub attenuation was found by taking the lowest of the two attenuations, $A_{\text{final}} = \min\{A(t), A(t + t_0)\}$.

The attenuation time series of the VSAT stations that were used in the final simulations used time series from separate days, acting as simultaneous time series but separated at such a distance as to be totally decorrelated. This assumption is justified since heavy rain is already somewhat decorrelated at a time separation of just 14.5 minutes, whereas a full 24 hour period would decorrelate the two time series to a much larger extent.

Chapter 6

Short-term Fade Prediction

6.1 Introduction

Following the implementation of a turbo-coding and site diversity scheme, several short-term fade predictors were investigated as an integral part of a general fade mitigation control scheme. Short-term predictors can be divided into three categories, namely (i) fade slope predictors which attempt to predict future samples based on the rate of change of attenuation over several previous samples, (ii) adaptive signal processing predictors, which use several previous samples with weights that constantly update depending on the attenuation dynamics and (iii) statistical predictors which predict future attenuation for a specified chance of exceedance.

In addition to the short-term predictors that are already available in the literature, two new predictors were developed. The first used an adaptive linear element (ADALINE), a basic component of neural networks, with a least-mean-squares (LMS) update method and the second used a hybrid signal processing predictor, for which both the ADALINE predictor and the auto-regressive moving-average (ARMA) predictor, proposed in [68], were constituent parts. The new predictors were compared with the predictors of other authors, the results of which appear in the later subsections of this chapter.

Finally, once all the predictors had been compared, a control strategy using the predictors was also investigated and is detailed in the last part of this chapter.

6.2 *Review of Short-term Predictors*

6.2.1 *Best-fit Slope Prediction*

A best-fit method of prediction was developed by Dossi in [66], where n previous samples are used to calculate the slope, $\zeta(t)$, of the first-order best-fit at time t . A prediction is then made with a forward prediction interval of k seconds as follows:

$$\hat{y}(t+k) = y(t) + k\zeta(t) \quad (6.1)$$

The Matlab code for this predictor appears in appendix I.

6.2.2 *Mean Slope Prediction*

A method of prediction was developed by Kastamonitis in [67] using the mean of the individual slopes over the previous n samples to obtain a value for $\zeta(t)$. A prediction is made with a forward prediction interval of k seconds using (6.1).

The Matlab code for this predictor appears in appendix J.

6.2.3 *ARMA Prediction*

An auto-regressive moving-average predictor was developed by Grémont in [121] where n previous samples were used to predict one sample ahead. A vector of coefficients, $\theta(t)$, and a vector of previous signal values and residuals, $\phi(t)$, were defined as:

$$\theta(t) = [a_1 \quad \dots \quad a_n \quad c_1 \quad \dots \quad c_n]^T \quad (6.2)$$

$$\varphi(t) = [-y(t-1) \quad \dots \quad -y(t-n_a) \quad \bar{e}(t-1) \quad \dots \quad \bar{e}(t-n_c)]^T \quad (6.3)$$

Single step ahead predictions were then made as follows:

$$\hat{y}(t) = \varphi^T(t) \theta(t) \quad (6.4)$$

And the coefficient vector, $\theta(t)$, was updated at each time step in the Gauss-Newton direction as follows:

$$\theta(t) = \theta(t-1) + \gamma R^{-1}(t) \psi(t) e(t) \quad (6.5)$$

where γ is a constant gain between zero and unity, $\psi(t)$ is a gradient vector w.r.t. θ and

$$R(N) = \frac{1}{N} \sum_{t=1}^N \varphi(t) \varphi^T(t) \quad (6.6)$$

However, in place of inverting R constantly, a computationally less intensive method described in [122] was used instead. This involved the substitution of equations 6.5 and 6.6 for a covariance matrix $P(t)$ and a prefilter $L(t)$ as follows.

$$L(t) = \frac{P(t-1) \psi(t)}{\lambda(t) + \psi^T(t) P(t-1) \psi(t)} \quad (6.7)$$

$$P(t) = \frac{1}{\lambda(t)} [P(t-1) - L(t) \psi^T(t) P(t-1)] \quad (6.8)$$

For a full description of this recursive least squares algorithm and an efficient method of inverting R , see [122]. The single step predictor is extended to a predictor of forward prediction interval, k , by transforming the a and c coefficients to f and g coefficients by an algorithm given in [123], and reproduced below.

$$F(q) = 1 + f_1 q^{-1} + \dots + f_{k-1} q^{-k+1} \quad (6.9)$$

and

$$G(q) = g_0 + g_1 q^{-1} + \dots + g_{n-1} q^{-n+1} \quad (6.10)$$

where $F(q)$ and $G(q)$ can be determined by equating the a and c coefficients of different powers of q^{-1} , provided $A(q)$ and $C(q)$ have zeros within the unit circle. The method of recursively updating coefficients does not guarantee this so the updated coefficients must be reflected into the unit circle to ensure stability. The calculation of f and g coefficients proceeds as follows:

$$\begin{aligned} c_1 &= a_1 + f_1 \\ c_2 &= a_2 + a_1 f_1 + f_2 \\ &\vdots \\ c_{k-1} &= a_{k-1} + a_{k-2} f_1 + a_{k-3} f_2 + \dots + a_1 f_{k-2} + f_{k-1} \\ c_k &= a_k + a_{k-1} f_1 + a_{k-2} f_2 + \dots + a_1 f_{k-1} + g_0 \\ c_{k+1} &= a_{k+1} + a_k f_1 + a_{k-1} f_2 + \dots + a_2 f_{k-1} + g_1 \\ &\vdots \\ c_n &= a_n + a_{n-1} f_1 + a_{n-2} f_2 + \dots + a_{n-k+1} f_{k-1} + g_{n-k} \\ 0 &= a_n f_1 + a_{n-1} f_2 + \dots + a_{n-k+2} f_{k-1} + g_{n-k+1} \\ &\vdots \\ 0 &= a_n f_{k-1} + g_{n-1} \end{aligned} \quad (6.11)$$

Chapter 6 – Short-term Fade Prediction

Modified vectors, θ' and φ' that contain the forward prediction interval values can now be defined, considering that now the prediction for the ARMA(3,3) model is made as follows:

$$\hat{y}(t+k) = \sum_{nc=1}^3 c_{nc} \hat{y}(t-nc) + y(t) + \sum_{ng=1}^3 g_{ng-1} y(t-ng) \quad (6.12)$$

and now

$$\theta' = [c_1 \quad \dots \quad c_n \quad g_1 \quad \dots \quad g_n]^T \quad (6.13)$$

and in addition to updating the residual we calculate the residual for the forward prediction interval, k , as

$$\hat{y}'(t) = \varphi'^T(t) \hat{\theta}'(t) \quad (6.14)$$

and update $\varphi'(t)$ as follows

$$\varphi'(t) = [-\hat{y}'(t-1) \quad \dots \quad -\hat{y}'(t-na) \quad y(t-1) \quad \dots \quad y(t-nc)]^T \quad (6.15)$$

In the original publication of this predictor [121], optimum values of n_a and n_c were found to be 3 in both cases. This was verified experimentally and was also found to be the case with the data being used for this comparison. It was not possible to find global optimum values of λ but values of 0.9860, 0.9998 and 0.9999 were found to give good results for the 20, 40 and 50 GHz time series respectively.

The Matlab code for this predictor appears in appendix K.

6.2.4 Statistical Prediction

A statistical predictor was defined in [117] where previous samples were used to predict the amount of resources that would not be exceeded by the attenuation for P percent of the time. For the purpose of comparison with the other predictors, an exceedance percentage of 50% has been used. A ten second moving-average is first applied:

$$\bar{y}(t) = \frac{1}{10} \sum_{i=0}^9 y(t-i) \quad (6.16)$$

and the prediction is made as follows:

$$\hat{y}(t+k) = m_A + \frac{2\sigma_A}{\pi} \operatorname{arcsinh} \left[\tan \left\{ \pi \left(0.5 - \frac{P}{100} \right) \right\} \right] \quad (6.17)$$

where:

$$m_A = \bar{y}(t) + \alpha_2 (\bar{y}(t) - \bar{y}(t-10)) \quad (6.18)$$

$$\sigma_A = \max(\bar{y}(t), 0) \sqrt{\beta_2 \Delta t} + \gamma_2 |\bar{y}(t) - \bar{y}(t-10)| \quad (6.19)$$

At $P = 50\%$ the second term drops out of the equation and the prediction reduces to $\hat{y}(t+k) = m_A$ out of equation (6.17). Values of the parameters γ_2 , β_2 and α_2 are provided in [117] for $k=1$ and $k=10$ second ahead predictions.

The Matlab code for this predictor appears in appendix L.

6.3 An ADALINE Predictor

A rain attenuation predictor was implemented using an adaptive linear element neural network with a single neuron and a number of previous samples, n , of 3 as shown in figure 6.1.

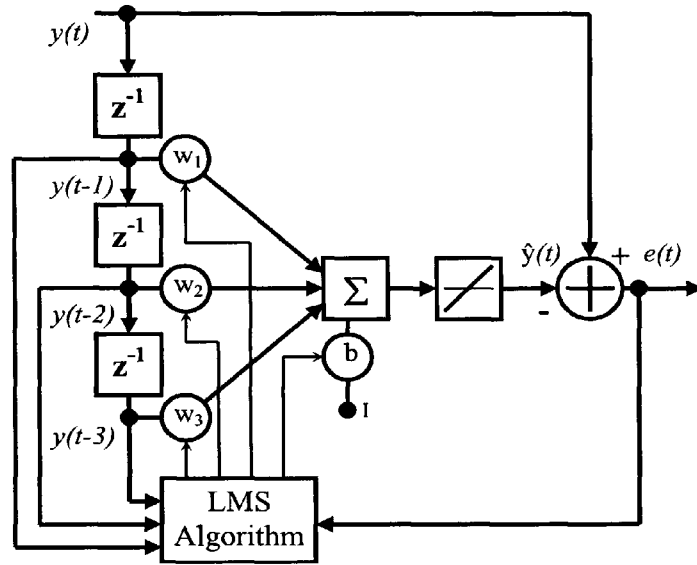


Figure 6.1: ADALINE fade predictor

The prediction $\hat{y}(t)$ is made as follows:

$$\hat{y}(t) = \mathbf{Y}_t^T \mathbf{W}_t + b \quad (6.20)$$

where b is the neuron bias, \mathbf{W}_t is the weight vector at time t

$$\mathbf{W}_t = [w_{1t} \quad \cdots \quad w_{nt}]^T \quad (6.21)$$

and \mathbf{Y}_t is a vector of previous inputs at time t

$$\mathbf{Y}_t = [y_{t-k} \quad \cdots \quad y_{t-k-n+1}]^T \quad (6.22)$$

Following each prediction, the weight vector and bias are updated by the LMS learning rule as follows:

$$\mathbf{W}_{t+1} = \mathbf{W}_t + 2\mu e_t \mathbf{Y}_t \quad (6.23)$$

$$b_{t+1} = b_t + 2\mu e_t \quad (6.24)$$

Chapter 6 – Short-term Fade Prediction

where μ is the learning rate that governs how quickly the weight vector is adapted to the changing input signal and e_t is the prediction error, i.e. the difference between the prediction made for the current time instant and the signal itself. It is necessary to select a value for μ that is sufficiently high to enable the predictor to track the event dynamics, but low enough to ensure stability. To aid the selection of this parameter, the full data set was segmented into 100-sample sections and analysed in terms of the maximum stable learning rate. This is equal to the reciprocal of the largest eigenvalue of the correlation matrix $\mathbf{Y}\mathbf{Y}^T$.

Using CDFs of maximum learning rate for the entire series, and testing the results for events whose dynamics changed very quickly to ensure stability and adequate tracking speed, learning rate values of 0.009, 0.0025 and 0.0015 were used for the 20, 40 and 50 GHz beacons respectively, which was the median value of learning rate. Lower values of learning rate, while no more stable during the generation of the final statistics, caused noticeable performance degradation.

An exhaustive discussion of the LMS algorithm with application to adaptive signal processing is given in [124].

Three methods of extending this predictor from a 1-step ahead to a k -step ahead predictor were compared. The first simply assumed that the prediction of prediction interval k was the same as the 1-step ahead prediction, i.e. a prediction is made from the previous 3 samples and compared with the current sample, the prediction weights are then updated according to the result. The second used a recursive method to predict one step ahead and then employed this prediction as the new sample to predict another

second ahead until a k -step prediction was reached. It should be noted that the weights were not updated by this recursive method. The final method was to simply put the predicted value k steps ahead and only update the weights when the signal could be compared k -steps later. This adds some initial delay, and also requires that the learning rates be reduced by a factor governed by the value of the prediction interval, k . Of these three methods the first gave the best results, indicating that the prediction error is large in comparison to the slow changing signal dynamics over these periods.

The Matlab code for this predictor appears in appendix M.

For the purpose of comparing predictors, two years of beacon data from the ITALSAT F1 and F2 satellites, measured by RAL at an earth station in Sparsholt, UK were analysed to isolate rain events.

Events were defined as starting where the 50 GHz beacon excess attenuation crossed a 1 dB threshold in the positive direction and ending where the 50 GHz beacon attenuation crossed a 1 dB threshold in the negative direction. Events were then selected by the following criteria:

- The full event without any bad or interpolated data was available for all three beacons, to allow a useful comparison between beacon frequencies if necessary.
 - The event duration exceeded 500 seconds and did not exceed 10000 seconds.
- These values were chosen to compare events that were long enough for a statistically significant comparison but not too large to create memory problems.

There are few events that last longer than 10000 seconds [125], so this should not impact too greatly on the variation of events selected.

- The maximum attenuation for the event reached at least 2 dB at 20 GHz. This is to ensure that some event dynamics are considered at all beacon frequencies, whereas if this condition were not included, some events at 40 and 50 GHz could be caused by cloud and would not cause any significant attenuation at 20 GHz.

In total, 111 events were selected that matched these criteria with a total duration of 207,004 seconds.

The overall predictor error is comprised of two components, an error caused by scintillation which can be considered uncorrelated at a sampling rate of 1 Hz [126] and Gaussian over periods of less than 1 minute [127], and an error made by the predictor of the underlying event dynamics. Figure 6.2(a) shows a typical stratiform event at 50 GHz and figure 6.2(b) shows the associated prediction error for the ADALINE predictor. The predictor error takes some initial period for convergence to occur since the event begins at 1 dB and the predictor begins predicting with zeros inserted in \mathbf{Y} .

Although a pre-filter can be used to remove scintillation, this adds a delay which is critical to the prediction result. Filtering can therefore be seen as a trade-off between effectiveness of scintillation removal and delay. Similar results occur if the number of previous samples used in the predictor, n , is increased. The three-sample predictor has been found in many cases to be optimum [66], [67], [121], as well as for the time series used in this analysis.

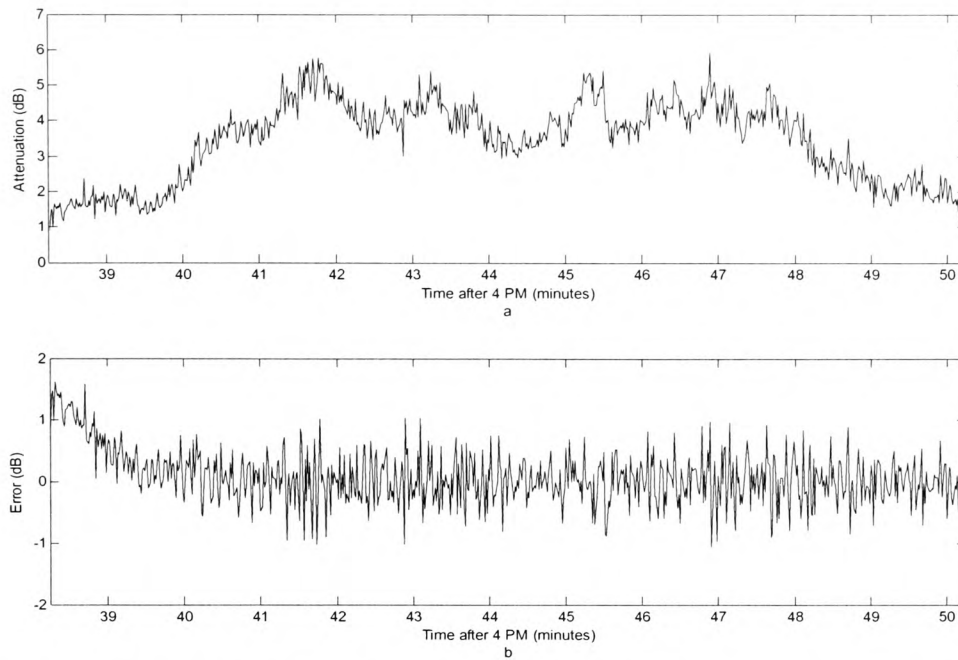


Figure 6.2: A rain event on the 5th November 1997 showing a) event attenuation and b) prediction error for the ADALINE predictor

Figures 6.3, 6.4 and 6.5 show the RMSE of the predictors for the 20, 40 and 50 GHz beacons respectively at all values of prediction interval k , with the exception of the statistical model for which model parameters have only been provided for 1 and 10 second ahead predictions [117]. Each predictor was implemented for the selected events from zero initial conditions for values of k from 1 to 20. The performance of each was evaluated in terms of the error distributions and the overall root mean square error (RMSE). It is noteworthy that the ADALINE predictor is at the worst disadvantage of all the prediction methods when zero initial conditions are used. This is because the events have been recorded after crossing the 1 dB threshold and thus the ADALINE predictor which has the slowest convergence rate is unfavourably biased by this initial error.

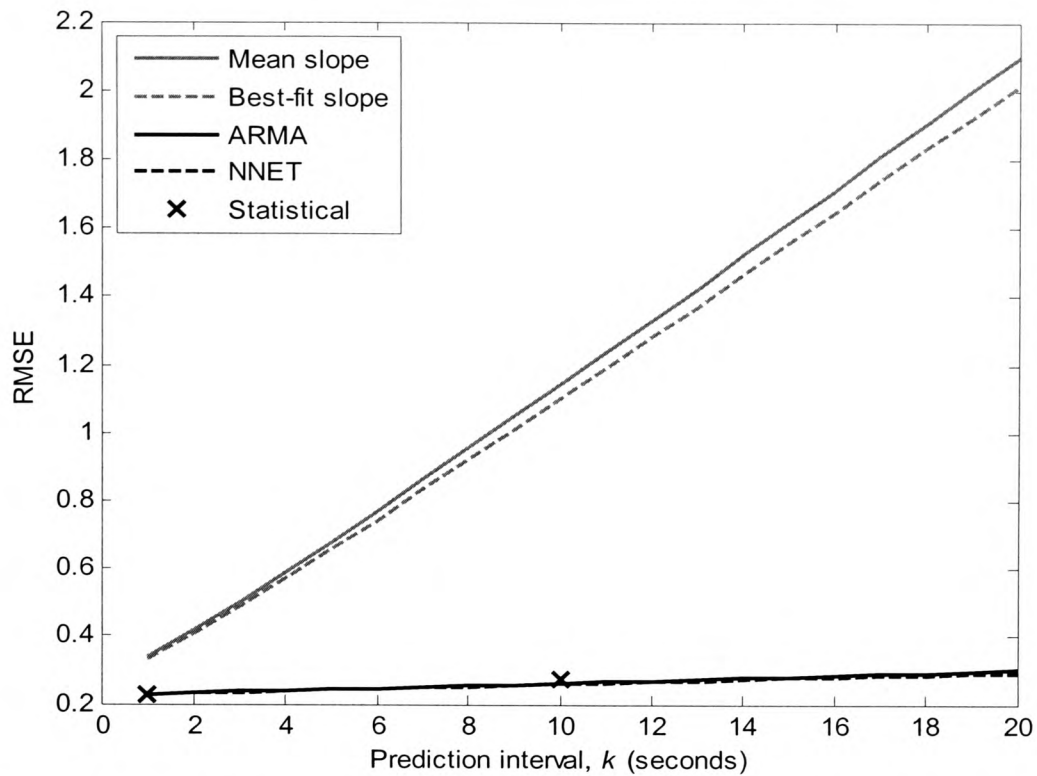


Figure 6.3: Root mean square error for each predictor for various prediction intervals at 20 GHz

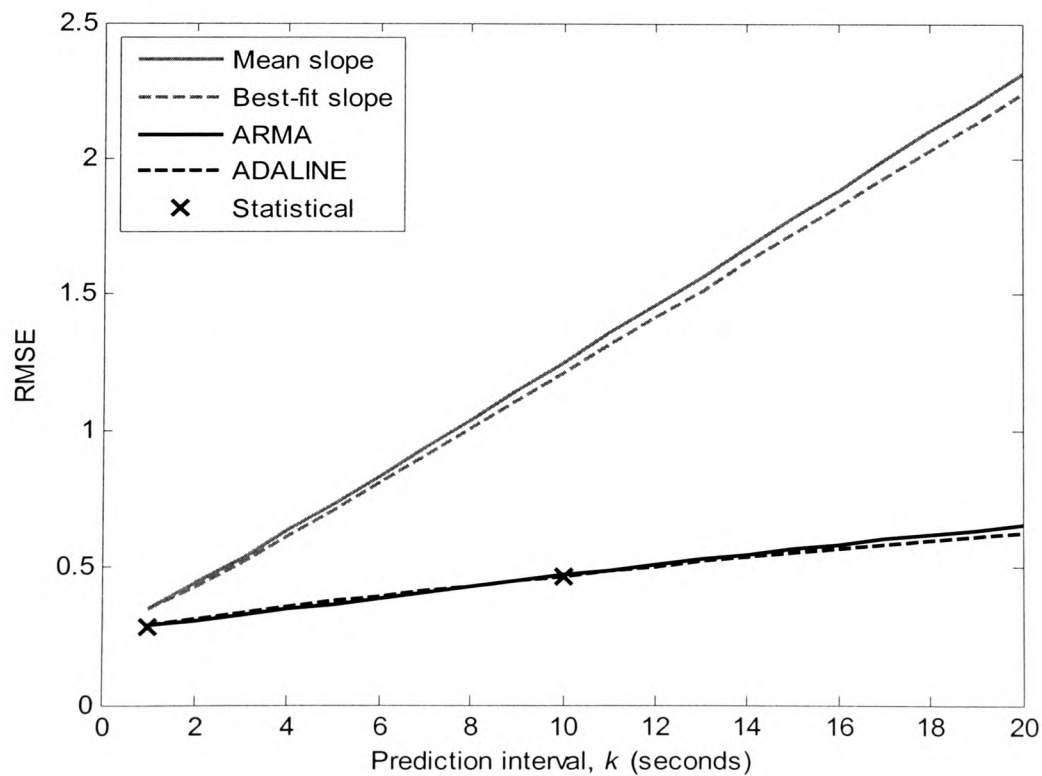


Figure 6.4: Root mean square error for each predictor for various prediction intervals at 40 GHz

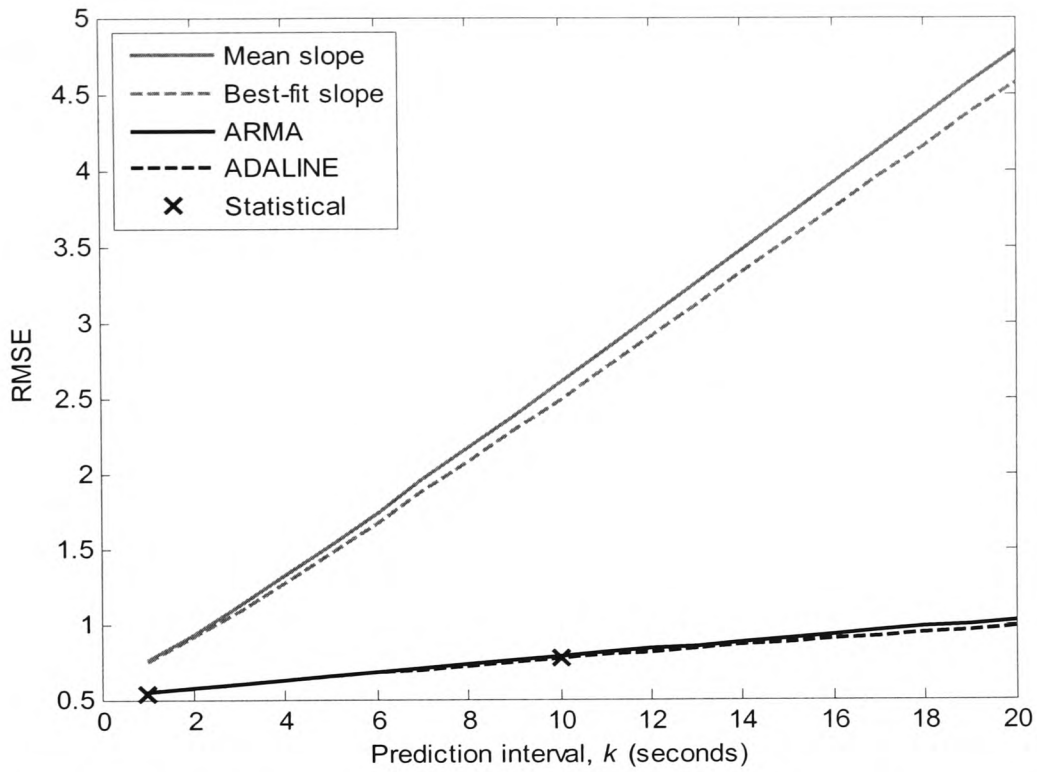


Figure 6.5: Root mean square error for each predictor for various prediction intervals at 50 GHz

In terms of the RMSE the ARMA, statistical and ADALINE predictors perform comparably well. The ARMA performs better than the ADALINE at lower values of prediction interval, whereas the ADALINE performs better at higher values of prediction interval for all beacon frequencies. This is partly because of the length of time it takes for the predictors to converge from zero initial conditions. When the predictors are compared with a short initial time period removed, such as in [19], so that only steady state conditions are compared, the ADALINE performs more favourably, even at low values of prediction interval. The statistical predictor performs comparably well with both the ARMA and ADALINE, with the exception of the 20 GHz beacon at $k = 10$, where it does less well.

Figures 6.6 to 6.8 show the cumulative distribution of under-predictions at prediction intervals of 1 and 10 for each predictor at 20, 40 and 50 GHz. This is a relative measure of the amount of extra resource required to prevent outages occurring for a given time percentage, and the results were very similar for all beacon frequencies. However, it is perhaps a less accurate reflection of the overall performance of each of the predictors since the initial offset causes the under-prediction to be biased in favour of the most rapidly converging methods. The RMSE considers both under-predictions and over-predictions and is therefore a better measure of the predictor's performance. Nevertheless, the CDFs are provided to give an indication of the amount of error that would have to be accounted for in any FMT switching control system.

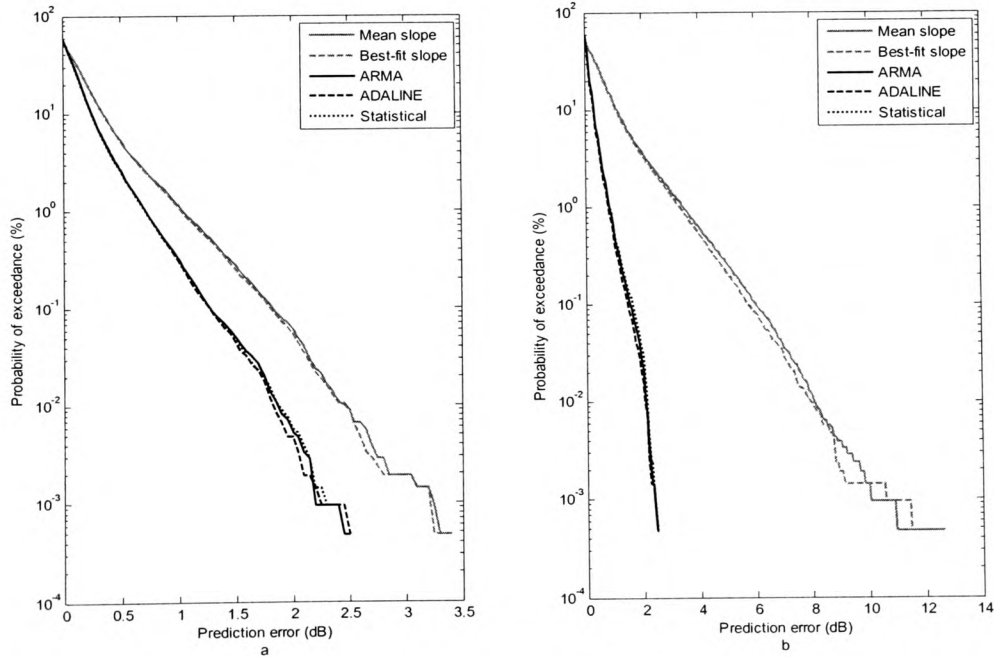


Figure 6.6: CDF of under-predictions at 20 GHz for prediction intervals of a) $k = 1$ and b) $k = 10$

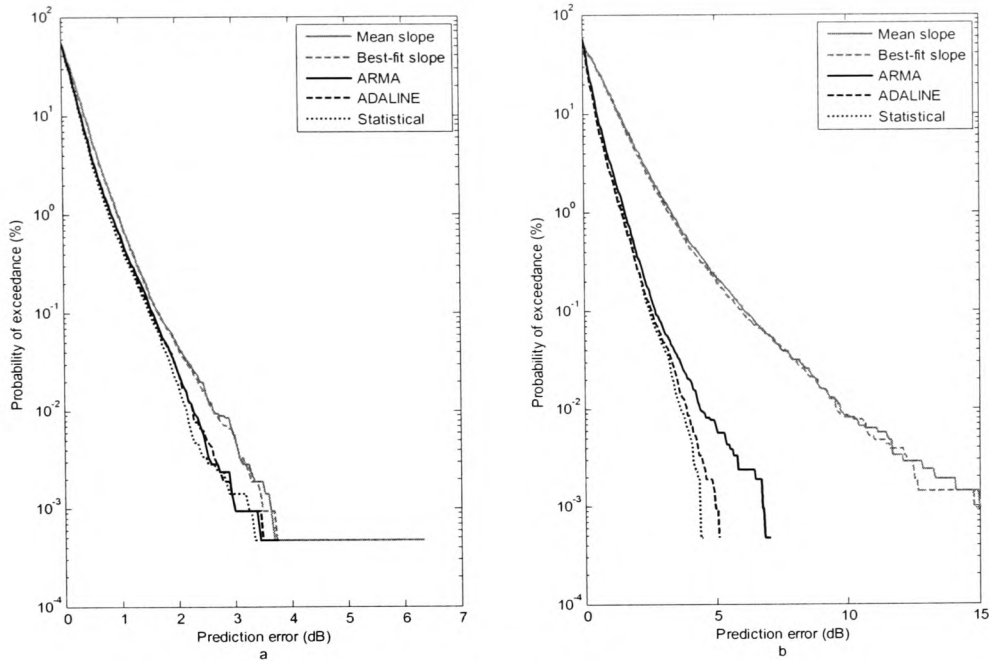


Figure 6.7: CDF of under-predictions at 40 GHz for prediction intervals of a) $k = 1$ and b) $k = 10$

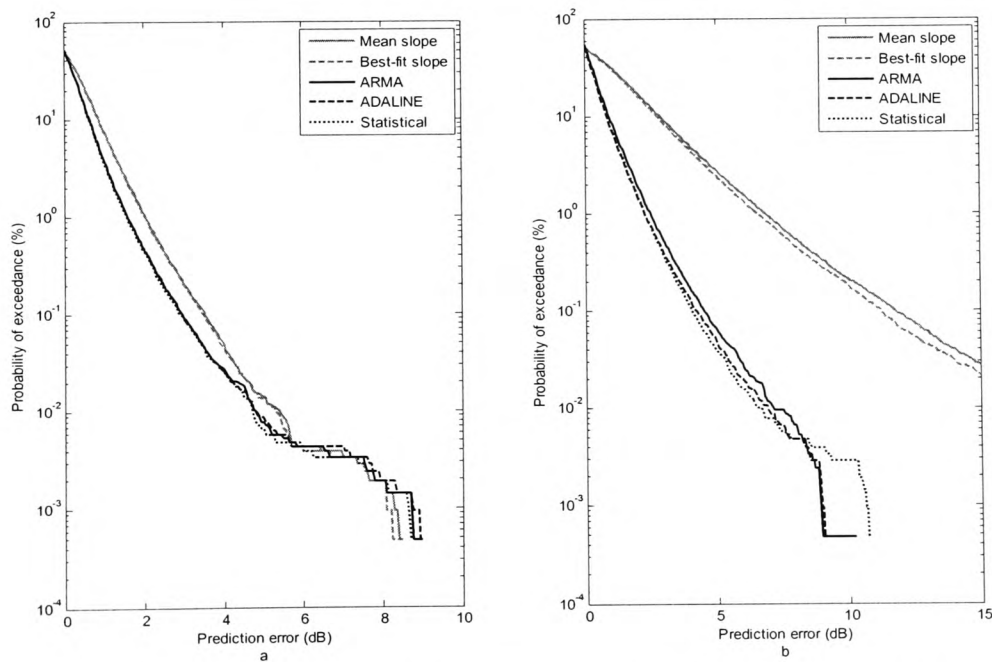


Figure 6.8: CDF of under-predictions at 50 GHz for prediction intervals of a) $k = 1$ and b) $k = 10$

Again, the signal processing and statistical predictors fare better than the fade slope methods, giving a system margin gain of approximately 1 dB for 99.99% certainty that there will be no under-prediction. Each of the signal processing and statistical methods

performs similarly. There is no significant difference for the 20 GHz time series but the statistical predictor performs marginally better at 99.99% certainty of no under-prediction for the 40 GHz time series and the ARMA and ADALINE performing marginally better at the same certainty of no under-prediction for the 50 GHz time series.

6.4 A Hybrid Signal Processing Predictor

Given that the ADALINE and ARMA predictors perform comparably well in [19], especially at low values of prediction interval, and since the Gauss-Newton update method of the ARMA predictor converges more quickly than the gradient method of the ADALINE predictor, the advantages of both methods can be exploited.

A simple selection criterion can be implemented to employ the predictor with the best recent performance at each sampling instant:

$$\varepsilon_c(t) = \gamma \varepsilon_c(t-1) + (y(t) - \hat{y}(t))^2 \quad (6.25)$$

Where $\varepsilon_c(t)$ is a measure of the current error performance of the predictor and γ is a decay factor between zero and unity which determines importance of previous predictions, the lower its value, the greater the bias of the most recent prediction. Whichever predictor has the lowest value of $\varepsilon_c(t)$ is selected to make the next prediction.

The Matlab code for this predictor appears in appendix N.

A subset of events of different types was used to determine the best decay factor, a value of 0.92 being found to give the best results in terms of the RMSE for this data. This changed very little for stratiform and convective rain events and can therefore be considered as a near-optimum value for southern England regardless of rain type.

Figures 6.9 and 6.10 show the events, predictor errors for both the ADALINE and ARMA predictors for a prediction interval of 1 second and the choices made by the hybrid selection procedure for a stratiform and convective event respectively.

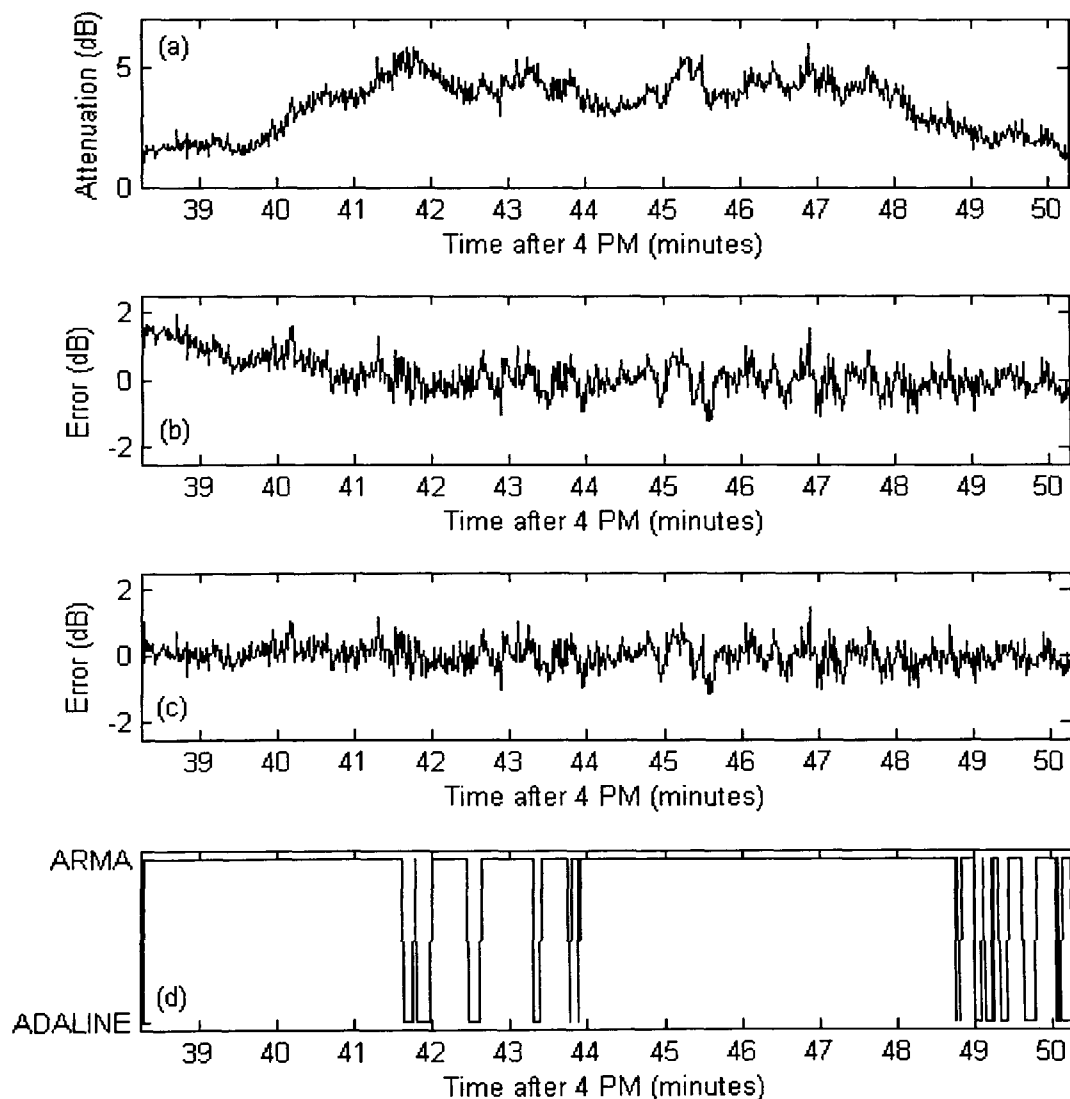


Figure 6.9: Stratiform event at 50 GHz on the 5th November 1997, showing a) Event attenuation, b) ADALINE predictor error, c) ARMA predictor error and d) the predictor choice

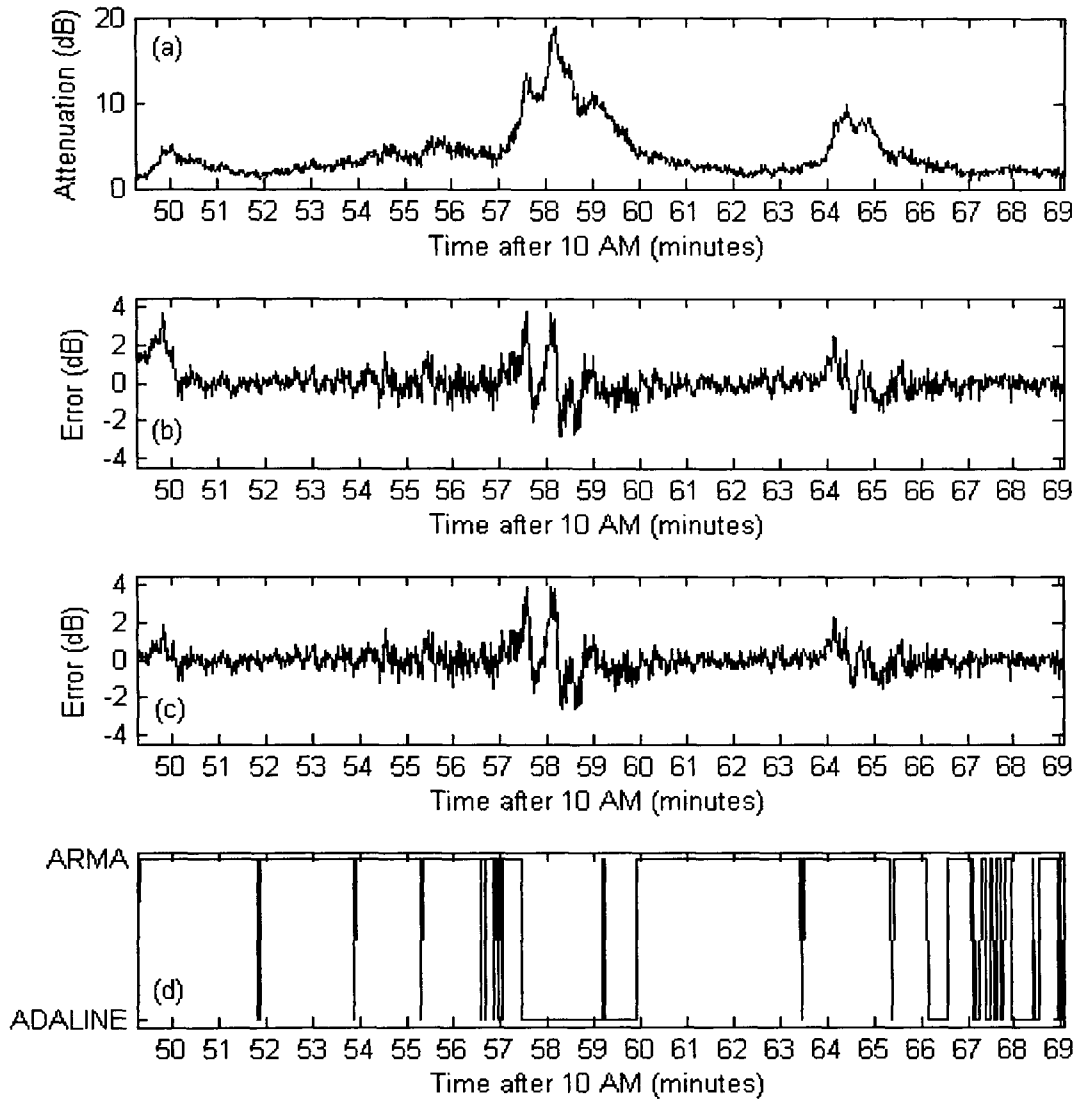


Figure 6.10: Convective event at 50 GHz on the 18th January 1998, showing a) Event attenuation, b) ADALINE predictor error, c) ARMA predictor error and d) the predictor choice

While for the stratiform event of figure 6.9 the ARMA predictor performs better for the majority of the time, for the event of figure 6.10 the ADALINE predictor performs better in the convective area between 57 and 60 minutes, which indicates that the hybrid may perform better than either of the two component predictors, since the different coefficient update methods are better suited to certain event dynamics. Since the Gauss-Newton update method attempts to converge in a single step, it is possible that it is attempting to adapt to the scintillation conditions rather than the underlying rain attenuation. This could be the reason that the ARMA predictor shows better

performance in slow-changing attenuation conditions as the scintillation is more effectively removed, but it may bias the predictor weights against the rain-attenuation. In other words there is a trade-off between two aspects of the predictor, its ability to remove scintillation and its ability to converge on the underlying event dynamics. The ARMA is better at the former, the ADALINE better at the latter.

The RMSE of the ARMA, ADALINE, statistical and hybrid predictors are shown in figures 6.11 to 6.13 for the 20, 40 and 50 GHz time series respectively. The hybrid predictor has a lower RMSE than either of the component predictors for all values of prediction interval and compares favourably with the statistical predictor at a prediction interval of $k = 10$.

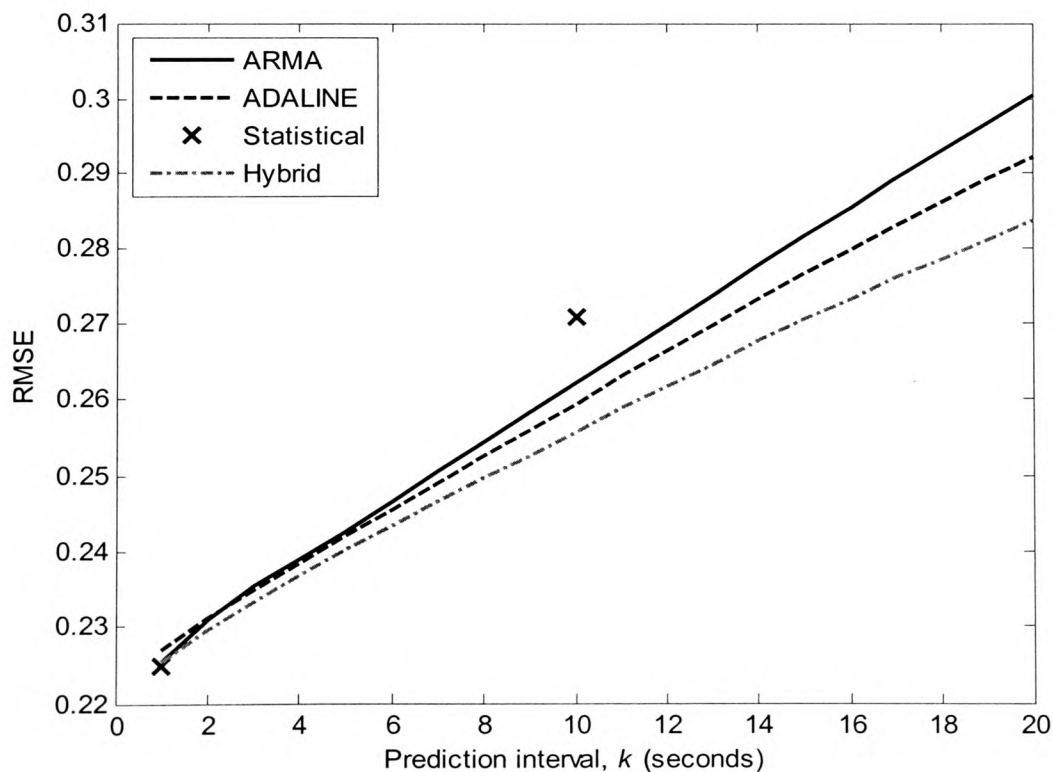


Figure 6.11: Root mean square error of hybrid predictor for various prediction intervals at 20 GHz

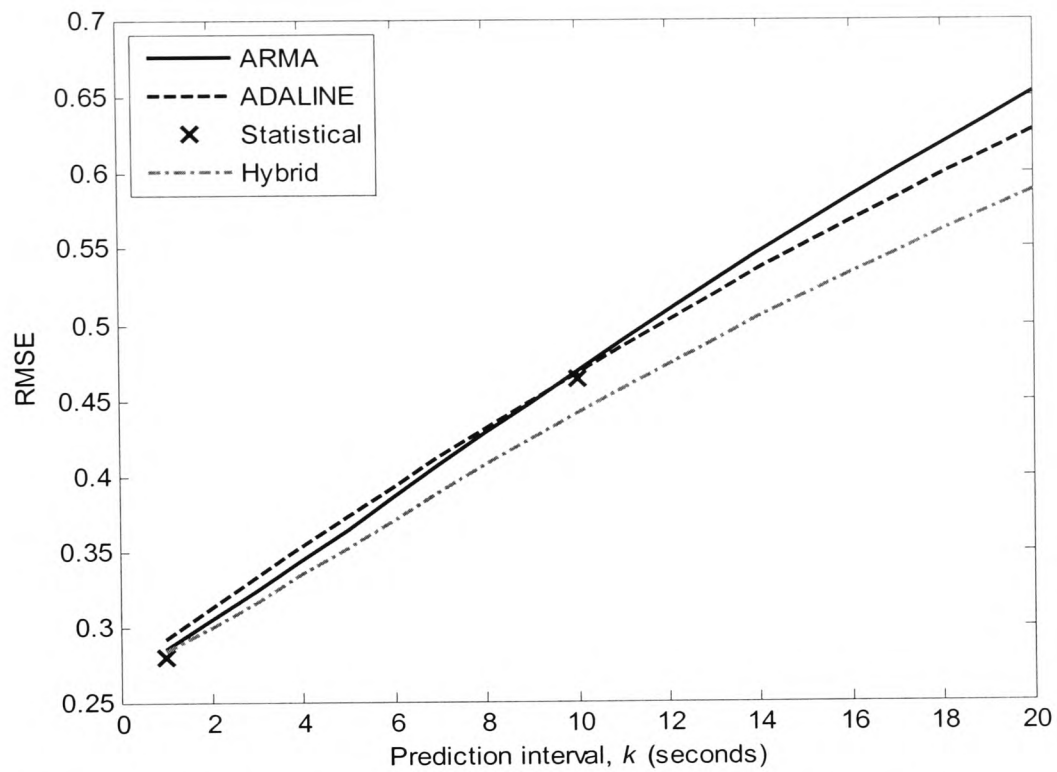


Figure 6.12: Root mean square error of hybrid predictor for various prediction intervals at 40 GHz

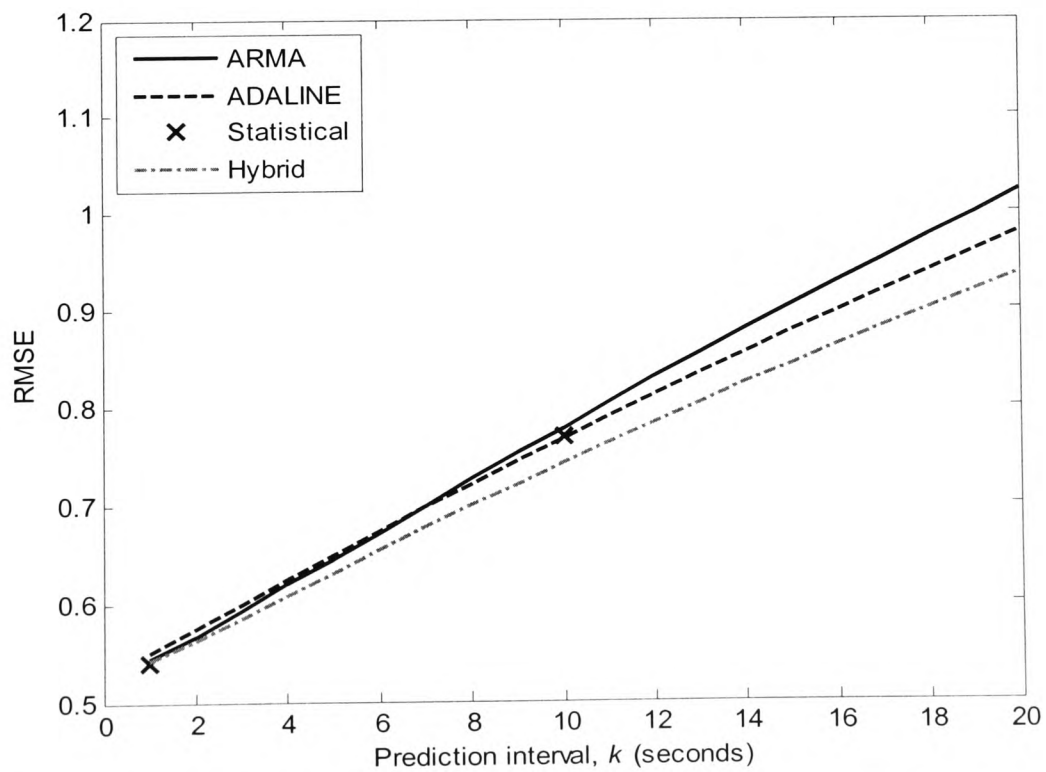


Figure 6.13: Root mean square error of hybrid predictor for various prediction intervals at 50 GHz

The CDFs of under-predictions for prediction intervals of 1 and 10 seconds are also provided in figures 6.14 to 6.16 for the 20, 40 and 50 GHz time series respectively. The gain is less noticeable in the CDFs as the performance increase occurs mainly in the upper region of the CDF, the outlying instances of large under-predictions performing only as well as the best of the two component predictors.

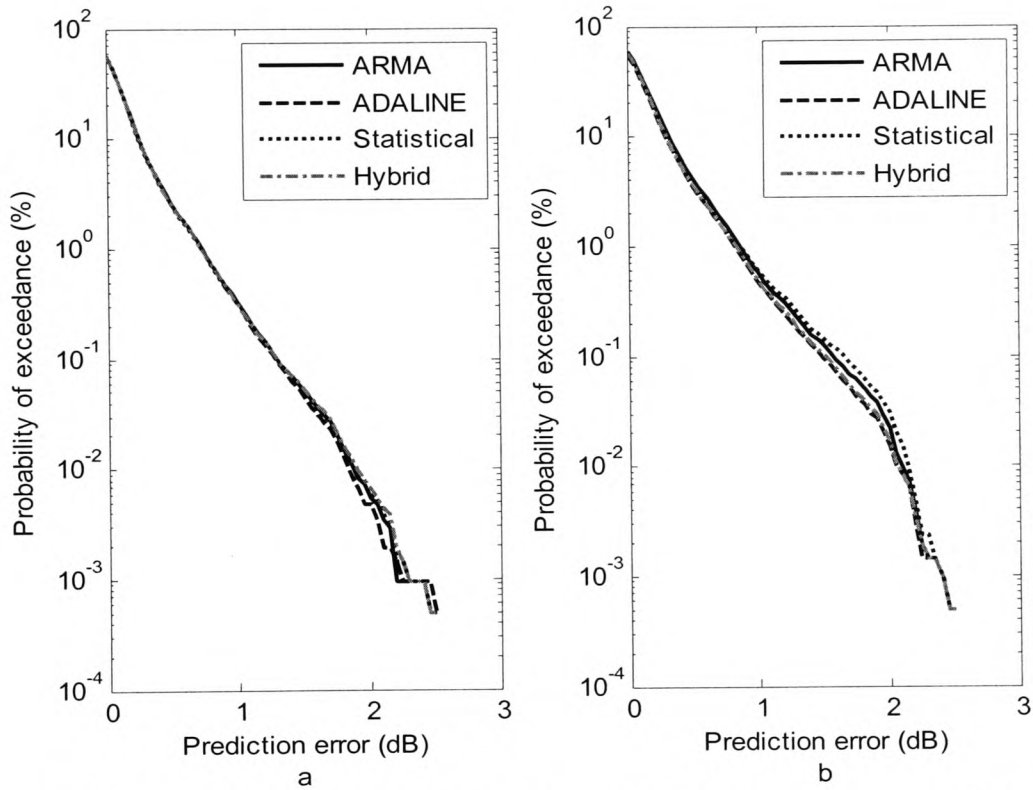


Figure 6.14: CDF of under-predictions for the hybrid predictor at 20 GHz for prediction intervals of a) $k = 1$ and b) $k = 10$

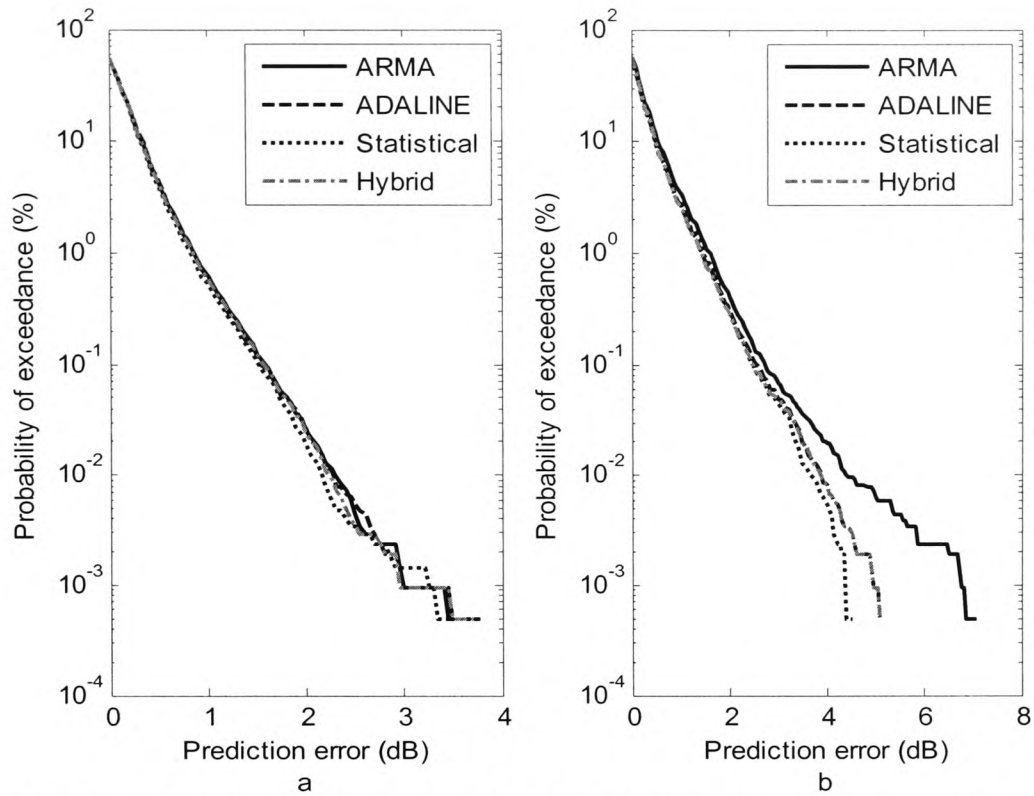


Figure 6.15: CDF of under-predictions for the hybrid predictor at 40 GHz for prediction intervals of a) $k = 1$ and b) $k = 10$

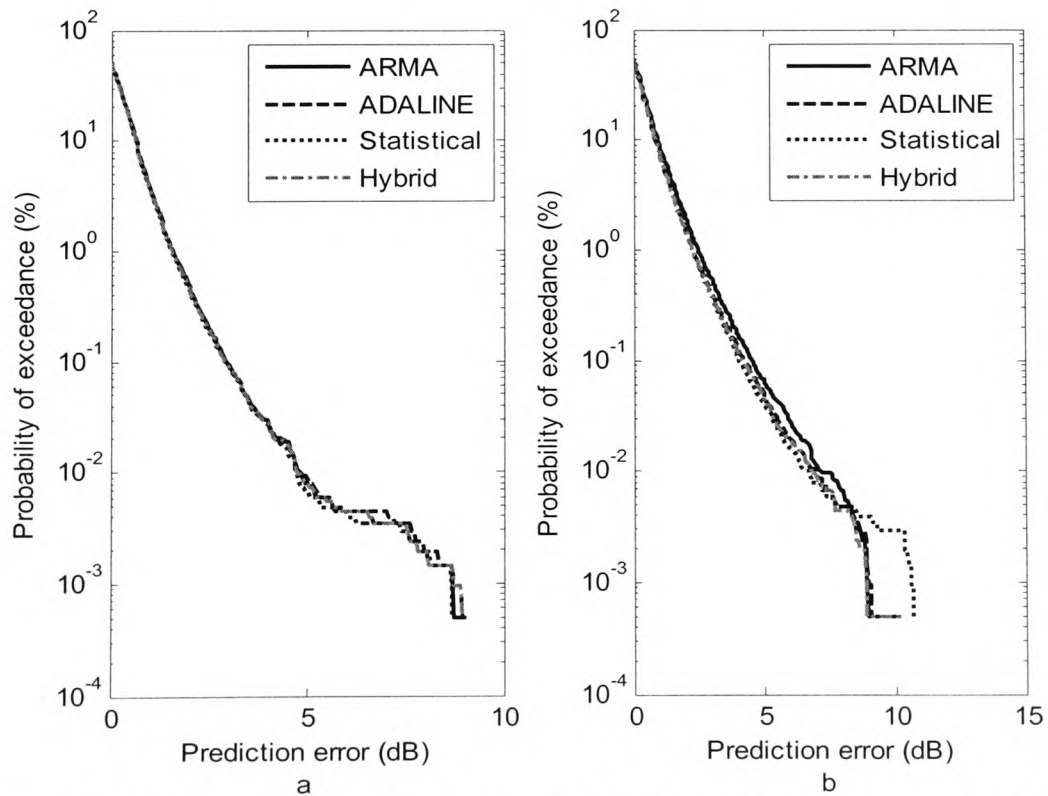


Figure 6.16: CDF of under-predictions for the hybrid predictor at 50 GHz for prediction intervals of a) $k = 1$ and b) $k = 10$

6.5 *Short-term Prediction Control*

A control mechanism based on consecutive forward predictions of the hybrid predictor discussed in the previous section was also investigated. Utilising the 111 events that were previously used to compare the prediction methods, switching based on the condition that all of K consecutive forward samples, crossed some upper or lower threshold was measured in terms of the percentage of threshold crossing and the percentage of switchovers to total samples.

For example, for $K = 1$, a 1 sample ahead prediction is made and must exceed a threshold value for a switchover to occur. However, at $K = 2$, a 2 sample ahead prediction is made and both that prediction and the previous prediction must exceed the threshold for a switchover to occur. Higher values of K , while less accurate in terms of prediction, are more resilient in terms of switchover which can be a useful property if switchovers require some overhead.

The first 200 seconds of an event from the 12th of July 2000 is shown in figure 6.17 in addition to the QoS threshold with a 1 dB scintillation margin for the cases $K = 1$, and $K = 10$. The effect of larger consecutive predictions, K is that fewer switchovers occur but in this scenario at the cost of more samples crossing the QoS threshold.

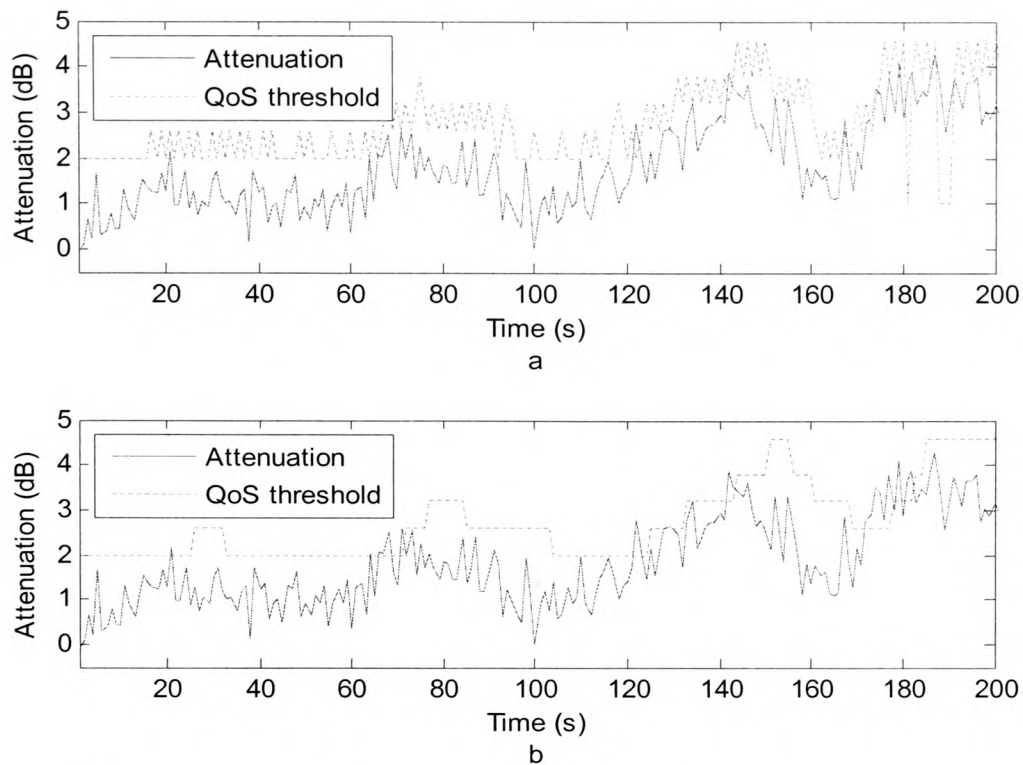


Figure 6.17: An event occurring on the 12th July 2000 showing a) the attenuation and QoS threshold where consecutive predictions, $K = 1$ and b) the attenuation and QoS threshold where consecutive predictions, $K = 10$

Further analysis was carried out in which the percentage of samples where a change in FMT parameters occurs and the percentage of samples for which the QoS threshold is crossed are compared for values of K from 1 to 10. The results appear in figure 6.18.

The results show that while the percentage of FMT switchovers continues to decline with increasing K , the percentage of samples where the signal crosses the QoS limit is a minimum at 3 to 5 samples. This behaviour can be accounted for by two different factors. Firstly, as K increases the chance of a switchover occurring decreases. This causes more samples to fall outside the QoS threshold as attenuation increases. However, for decreasing attenuation, fewer samples will be caused to fall outside the QoS threshold due to scintillation as the FMT level will not be reduced as quickly.

Secondly, the prediction error increases as K increases and this also contributes to an increasing number of QoS threshold crossings with increasing K .

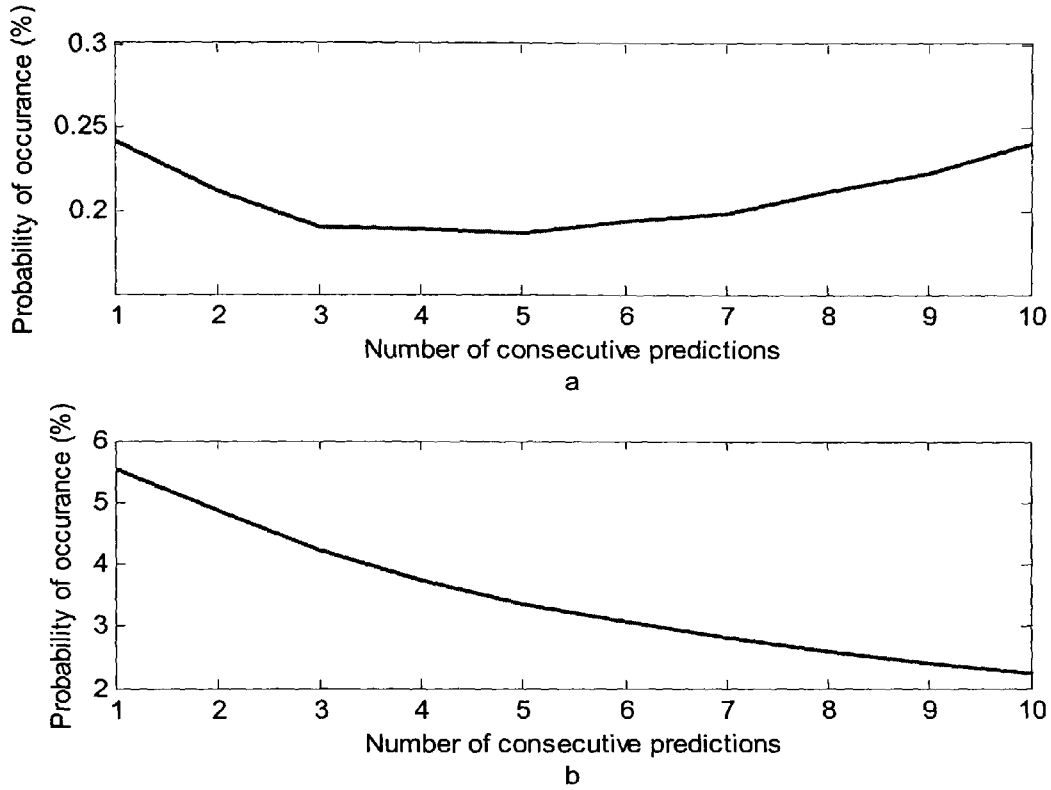


Figure 6.18: Performance of short-term prediction control showing a) the percentage of samples for which the QoS threshold is crossed and b) the percentage of samples for which a change in FMT parameters occurs

6.6 Conclusions

The simulations carried out in chapters 8 and 9 utilise this analysis in predicting future samples of attenuation and switching between FMT types according to the fading level. A four-sample value of K was used throughout as figure 6.18 shows this to be a good trade-off between QoS performance and switching efficiency. However, in this work the ADALINE predictor has been used due to practical considerations with the code

Chapter 6 – Short-term Fade Prediction

execution. The use of the hybrid predictor would deliver marginally better results, but would have increased the simulation time enough to make the simulations impractical.

Chapter 7

Preliminary System Design and Pre-processing

7.1 Introduction

The previous chapters have detailed some of the elements that are necessary in a two way V band satellite link for broadband internet delivery. However, several elements remain before a system can be simulated and this chapter aims to explain these elements and the approach taken to include them in the simulations in the following chapters. It should be noted that while all of this information is utilised in the final simulations (chapter 9), not all of it is utilised in the preliminary simulations (chapter 8). A more detailed explanation of which parts are applicable in each case appears in both these chapters.

In brief, the remaining elements to be considered are:

- Pre-processing of the raw data time series to create a full time series without data gaps and to produce frequency scaled time series for implementation of frequency diversity.
- Consideration of the simulation scenario and the traffic generated at each of the VSATs.
- A dual-channel protocol for MF-TDMA multiple access, suitable for passing control information between VSAT and hub.

Each of these elements will be covered in the following sections of this chapter.

7.2 *Data Pre-processing*

In order for a V band system simulation to take place, a full year of attenuation measurements taken at RAL were analysed and pre-processed to yield a time series of attenuation that could provide a real rain-fading scenario to the simulation.

The Radio Communications Research Unit at CCLRC-Rutherford Appleton Laboratory conducted an experiment involving the ITALSAT satellites from December 1995 up until the end of ITALSAT F2's operational lifetime in July 2002. The majority of the data was collected from the beacons carried on the ITALSAT F1 satellite at frequencies near to 20, 40 and 50 GHz, though some 20 GHz data was collected from ITALSAT F2 in the later years of the experiment. A tracking system was employed with the receivers operating near 40 and 50 GHz to compensate for the North/South movement of the ITALSAT F1 satellite; this helped to preserve the receivers' dynamic range of ~20 dB for observation of signal fading. The characteristics of the beacon receiving system are given in [77] and [128].

The raw data recorded by the beacon receivers was sampled at a rate of 1 Hz. Due to the diurnal variation of the satellite, slow variation from day to day in the signal levels as a result of thermal shifts and resultant power supply fluctuations, and sudden changes in signal levels due to satellite manoeuvres, preprocessing was required to produce accurate time series of attenuation from the recorded power levels. Attenuation time series were produced relative to vacuum (i.e. total attenuation) and relative to clear sky (i.e. excess attenuation, which excludes contributions from atmospheric gases). The gaseous attenuation and sky brightness temperature at each of the beacon frequencies

were estimated from ground measurements of temperature, pressure and relative humidity [129] [130] and the clear-sky level was calibrated by a co-located radiometer operating at 51 GHz. Finally, a zero dB reference level was established for each of the beacon frequencies in 10 minute intervals by establishing clear-sky points from the sky brightness temperature and using a Fourier series fitting method such as that in [86] to improve the clear-sky estimates, given that large gaps with no clear-sky can occur. All of these aspects of the data pre-processing were carried out at RAL, resulting in attenuation time series for total attenuation and excess attenuation.

The resulting data was also analyzed by RAL and a flag provided for each sample indicating the validity of the data, see table 7.1.

Flag	Data Validity
0	Not Valid
1	Valid
2	Attenuation not genuine but data valid
3	Loss of lock, data interpolated from another frequency
5	Loss of lock, no interpolation possible

Table 7.1: Data flag interpretation

Pre-processing of the attenuation time series involved checking each series for errors in the data flag and taking appropriate action depending on the circumstances. The analysis was carried out using Matlab in two stages. The first simply identified days that were suitable and the second performed certain scaling and interpolation as appropriate on the selected days.

The general coding flow diagram for the first stage appears in figure 7.1. The process involves opening the time series for each day if it exists and checking for gaps of

missing data or anomalies between the data and the data flag. If no anomalies are present and no gaps greater than 300 seconds are present, the time series is flagged as good. This is done for all three available beacons and a final check is made to determine whether gaps greater than 300 seconds in any one time series can be frequency scaled from another frequency. If so the day is added to a record of useful days.

The Matlab code for this stage appears in appendix O.

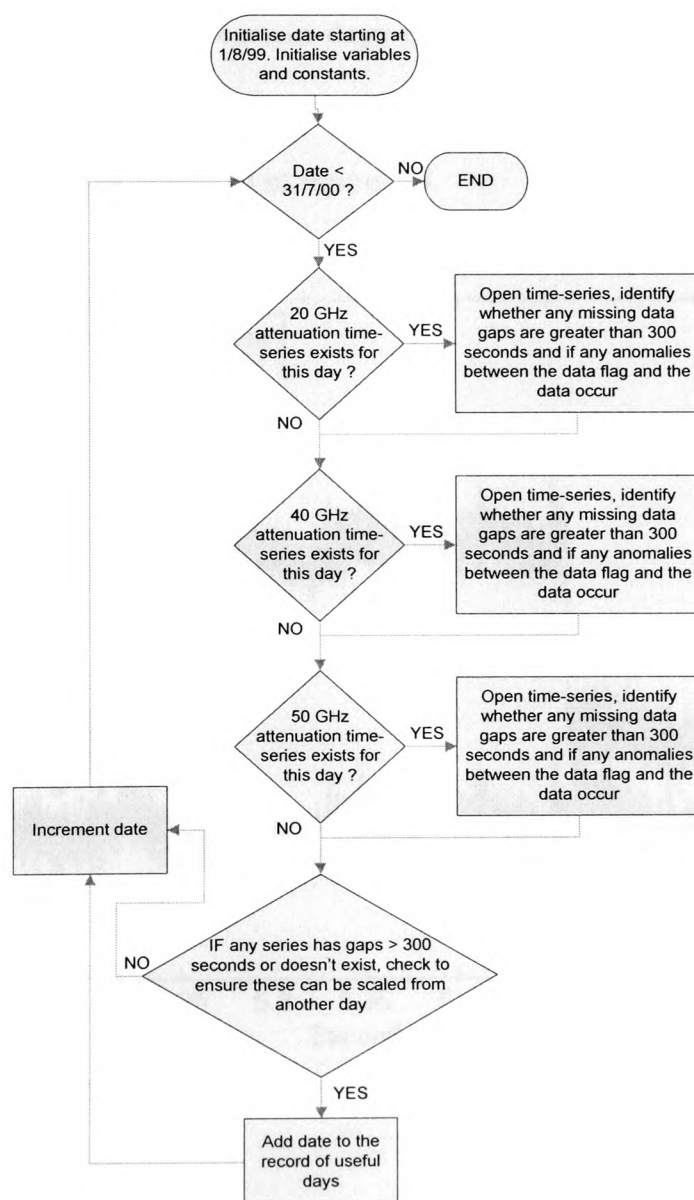


Figure 7.1: Flow diagram for software that selects which days are useful

Several days showed strange behaviour where the data flag indicated that the data was valid but from visual inspection it was obvious that this was not the case. These anomalies fell into three categories. Firstly, on certain days the total attenuation rose suddenly over a single sample but the excess attenuation did not, later returning to a reasonable value with a sudden fall. An example of this is shown in figure 7.2. Secondly, for certain days where attenuation during a heavy rainstorm occurred, the attenuation at the point where the receiver lost phase lock increased to very large values over a single sample for both total and excess attenuation, an example of which is shown in figure 7.3. Lastly, for certain days the attenuation dropped from zero to a significant negative value (i.e. a gain) for a few hundred seconds before returning to normal. An example of this is shown in figure 7.4.

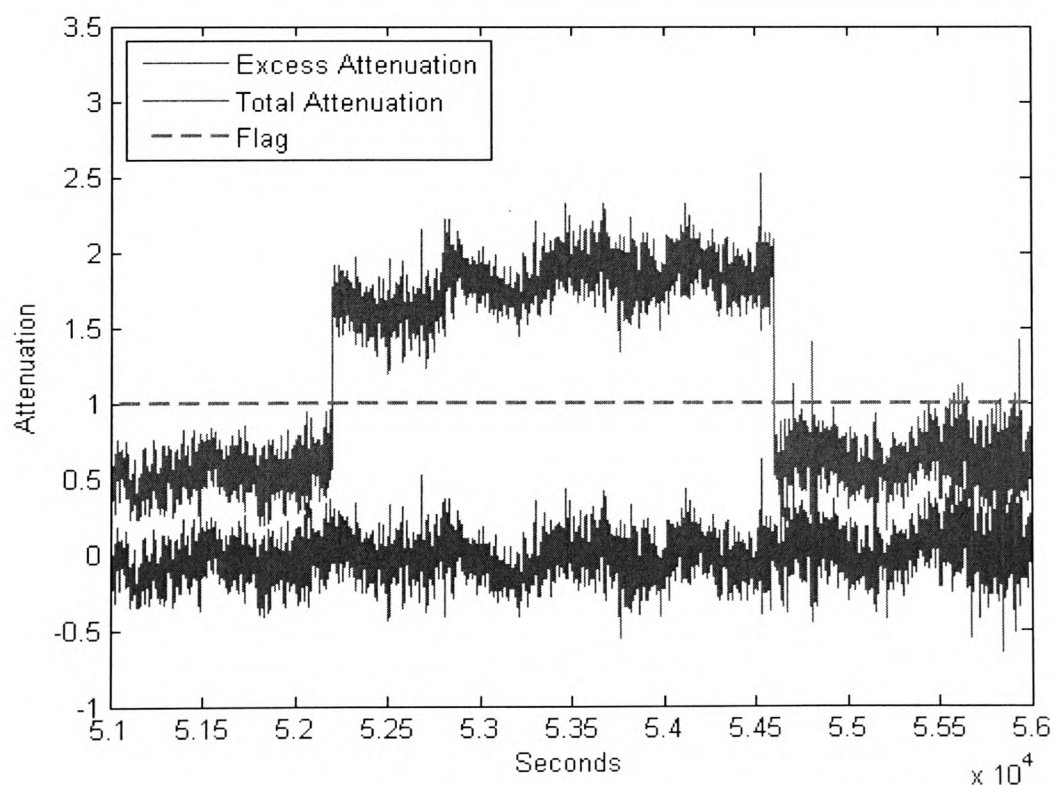


Figure 7.2: Data anomaly of the first type at 20 GHz

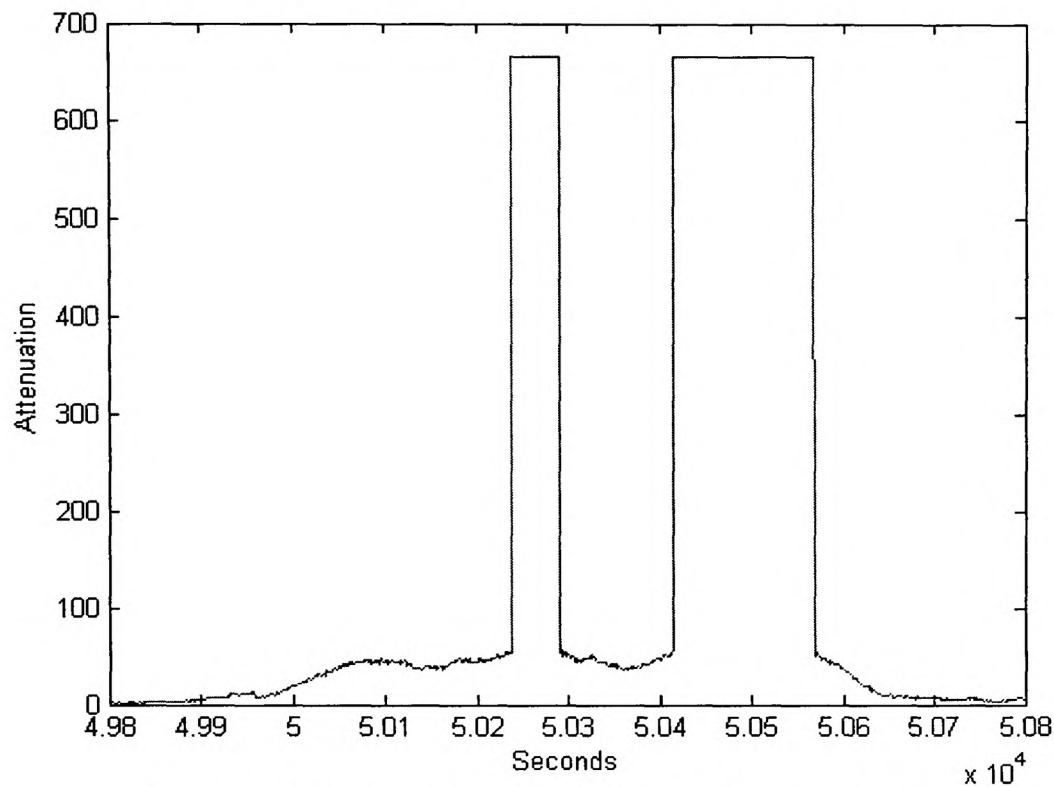


Figure 7.3: Data anomaly of the second type at 40 GHz

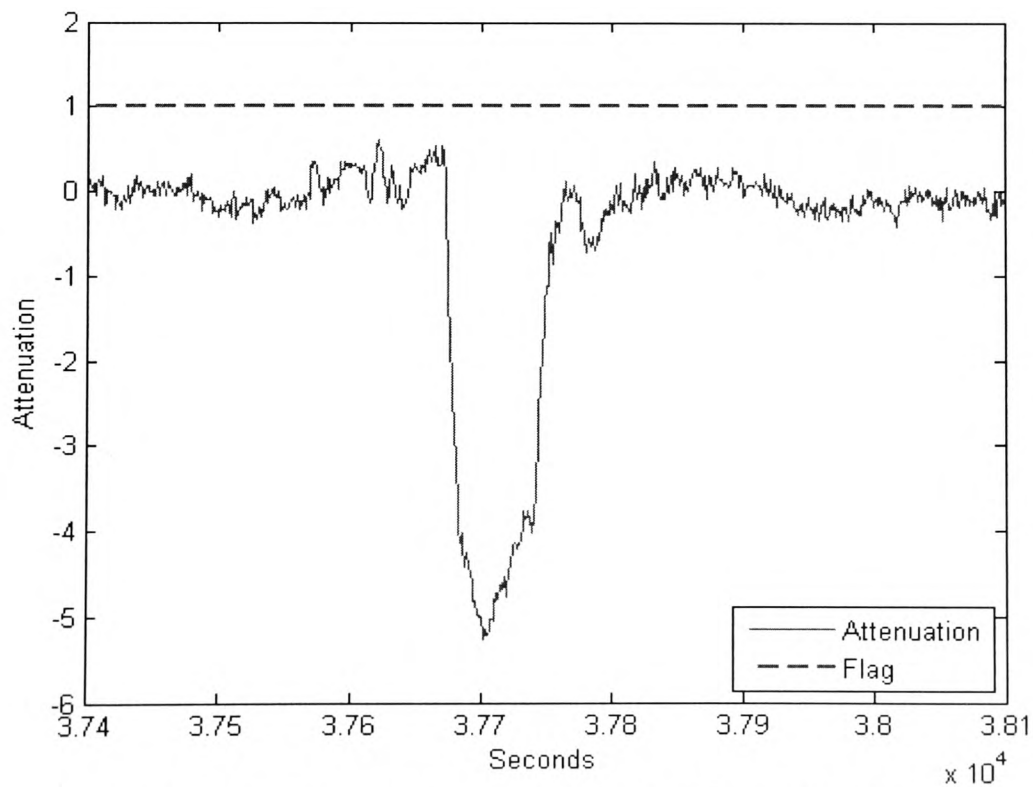


Figure 7.4: Data anomaly of the third type at 20 GHz

Chapter 7 – Preliminary System Design and Pre-processing

Days in which these anomalies occurred were removed from the set of valid days in order to ensure an accurate set of data before the second pre-processing stage was carried out.

The coding flow diagram for the second stage appears in figure 7.5. This program takes the list of good days that was compiled by the previous Matlab program and either scales data gaps from time series data at another frequency or interpolates gaps if this is not available. Scaling of data gaps from a time series of one frequency to another was carried out in the same manner regardless of whether the entire time series was frequency scaled or just a small section of bad data. The scaling was carried out in the following manner:

1. A rain attenuation time series and scintillation time series were generated immediately as each time series was opened. The rain attenuation time series was generated by an $N=40$ FIR filter designed in Matlab with a cut-off frequency of 20 mHz. The resulting filtered time series was also shifted to remove the filtering delay. The scintillation time series was simply the original time series minus the filtered version. Finally a time series of gaseous attenuation was created by subtracting the excess attenuation from the total attenuation for the original time series.
2. A new time series for rain attenuation was created for the missing data. This was done with the ITU-R model for frequency scaling [9], scaling from the available frequency to the unavailable frequency.
3. A new time series for scintillation was created for the missing data. This was done with the frequency dependent equations and constants for scintillation

intensity in [131], also scaling from the available frequency to the unavailable frequency.

4. The new time series for rain attenuation and scintillation intensity were added together to give an excess attenuation time series.
5. A new gaseous attenuation time series was created from the gaseous attenuation time series at the available frequency by scaling. The scaling was done according to the statistical properties of gaseous attenuation for the same location, which is given in [125]. The gaseous attenuation variance was first scaled up or down (multiplied or divided) depending on whether the scaling was going from a lower to higher frequency or a higher to lower frequency respectively. The scaling constants were 1.1, 1.35 and 1.485 for the 20 GHz to 40 GHz, 40 GHz to 50 GHz and 20 GHz to 50 GHz operations respectively. This was followed by adding or subtracting a constant factor that corresponded to the mean difference in gaseous attenuation between the two frequencies. These values were 0.4 dB, 1.9 dB and 2.3 dB for the 20 GHz to 40 GHz, 40 GHz to 50 GHz and 20 GHz to 50 GHz operations respectively.
6. The new gaseous attenuation time series was added to the excess attenuation time series that had already been generated to yield a total attenuation time series at the unavailable frequency.

The Matlab code for this stage appears in appendix P.

Chapter 7 – Preliminary System Design and Pre-processing

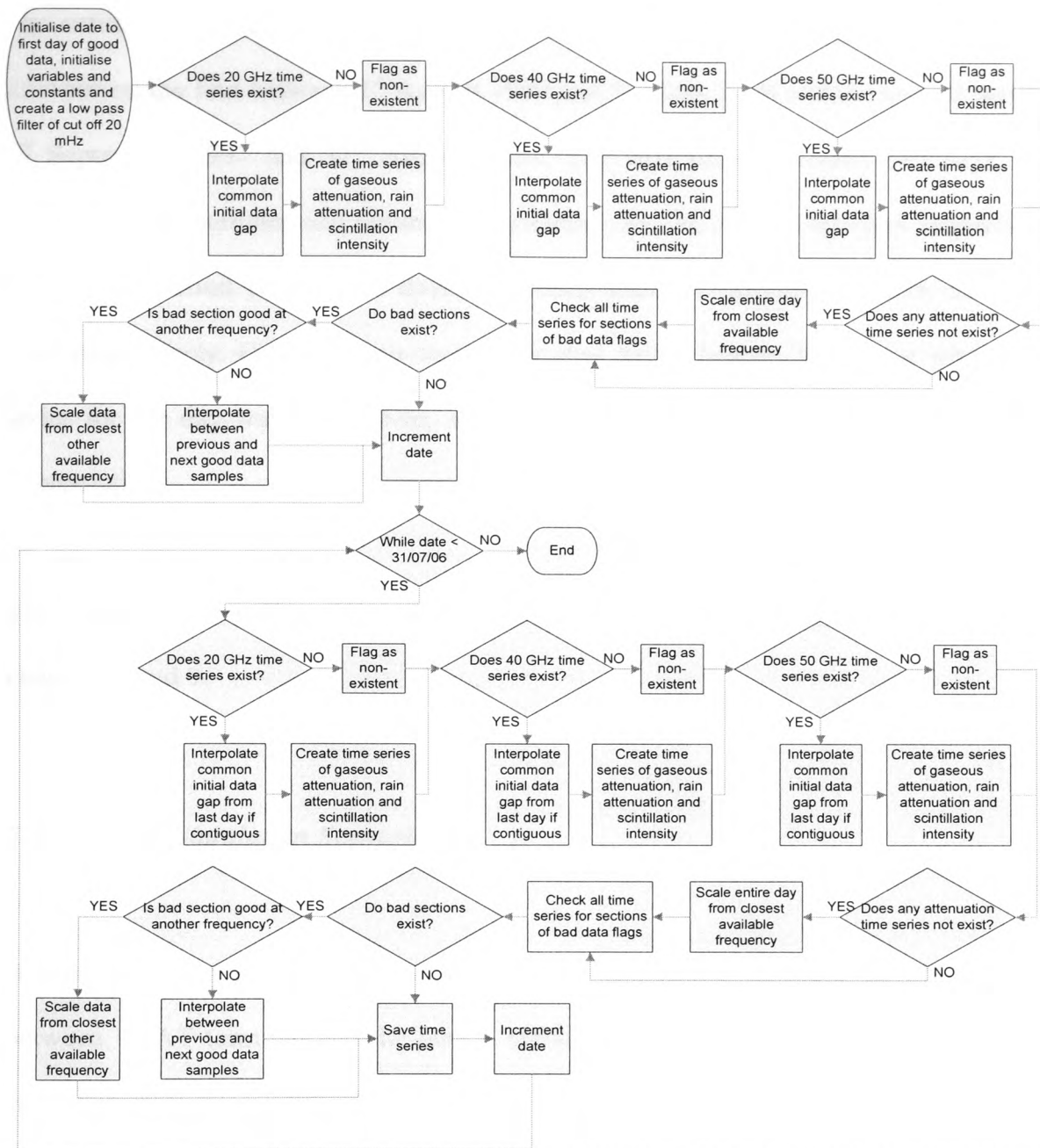


Figure 7.5: Flow diagram for software that scales or interpolates bad data sections

In the event that data gaps exist that cannot be entirely scaled from a time series at another frequency, provided the gaps are not longer than the chosen threshold of 300 seconds which was identified in the first stage, the gap is filled with a straightforward linear interpolation between the last good data point and the next good data point.

Chapter 7 – Preliminary System Design and Pre-processing

The entire procedure described above was applied to two separate years of attenuation time series, the first between 1st August 1999 and 31st July 2000 and the second between 1st September 1997 and 31st August 1998. The first period yielded 249 days, with winter, spring, summer and autumn contributing 55, 70, 55 and 69 days respectively. The second period yielded 193 days, with each season contributing 33, 55, 52 and 53 days respectively. Omitted days were ones that had a beacon frequency whose time series for the day was incomplete, and the gap could not be filled through scaling or interpolation.

The time series resulting from this procedure was used in the system simulations of chapters 8 and 9.

7.3 *System Simulation Scenario*

Before designing a protocol that is able to communicate changes in FMT scheme between VSAT terminal and the hub, a scenario must first be proposed. The scenario chosen in this case is one of 620 VSAT terminals across Europe that are assumed to have large separation distances and therefore have no significant correlation of rain attenuation. This assumption simplifies the creation of attenuation time series but should be close to the actual case since satellite internet provision is likely to be more expensive than the equivalent terrestrial services, making it desirable only to those who cannot otherwise receive terrestrial services. This is likely to include home users in remote areas, rather than large clusters of users in urban areas. A value of 620 VSATs was chosen by an iterative process so that not more than 58 stations were active for more than 0.01% of the time.

Chapter 7 – Preliminary System Design and Pre-processing

The stations are all assumed to have an asymmetrical bandwidth allocation of 512 kbps on the forward link and 64 kbps on the return link. This asymmetry is typical of most home user broadband connections where download capacity is more heavily utilised than upload capacity. These bit rates were chosen as a low bandwidth broadband system, although fewer stations could be chosen at a higher bandwidth per station without any difficulty. At the link budget stage of the work, these bit rates were used to determine the overall burst rate of the different schemes for the traffic channel (see the protocol in the following section).

Although the main focus of this work is to compare the performance of services at V band with various FMT schemes and those provided via a fixed margin, some simulation of traffic is useful, if only to put the comparison in a more realistic context. For this purpose, it is assumed that the satellite connections are of a dial-up nature and the traffic is dealt with as though the terminals were making calls with exponential inter-arrival times, i.e. a Poisson model. Although internet traffic has been shown to be self-similar in nature [132], the Poisson model is a good first approximation and is sufficient for the purpose of this work. However, this assumption gives an optimistic scenario in terms of an Internet traffic model, but the inclusion of more complex models would require extra processing which would lengthen execution time even further and was therefore avoided. Table 7.2 shows the statistics for home internet usage in January 2007 taken from [133].

Calls (web sessions)	35/month
Domains visited	73/month
Web pages	1547/month
Page views per surfing session	43
Average usage time	32h 18m 31s /month
Time spent during call	55m 37s
Average webpage viewing duration	45 seconds

Table 7.2: Internet usage statistics

From the calls per month, and time spent during call, the average inter-call duration and the average call duration were calculated to be 75,086 seconds and 3337 seconds respectively. However, while creating the traffic, the average inter-call duration was reduced to the limit of Poisson-like behaviour in order to take a pessimistic view of the amount of traffic. The final value of inter-call duration used was 60,000 seconds, or roughly 44 sessions per month. When the inter-call duration becomes shorter than this, the concurrent calls begin to exhibit non-Poisson behaviour as the assumption of short call duration compared with inter-call duration becomes increasingly less accurate.

Two matrices were generated in Matlab, one of which contained the start times, the other containing the finishing times of activity for each of the 620 stations. This was achieved by generating a random exponential number, for the duration of a call and for the time period between calls for each station, using the appropriate average value, having initialised each station by 20 days of activity beforehand to ensure that the results were not affected by starting conditions.

The resulting PDF showing the distribution of number of active stations appears in figure 7.6. The results give a good visual fit with the expected Poisson distribution.

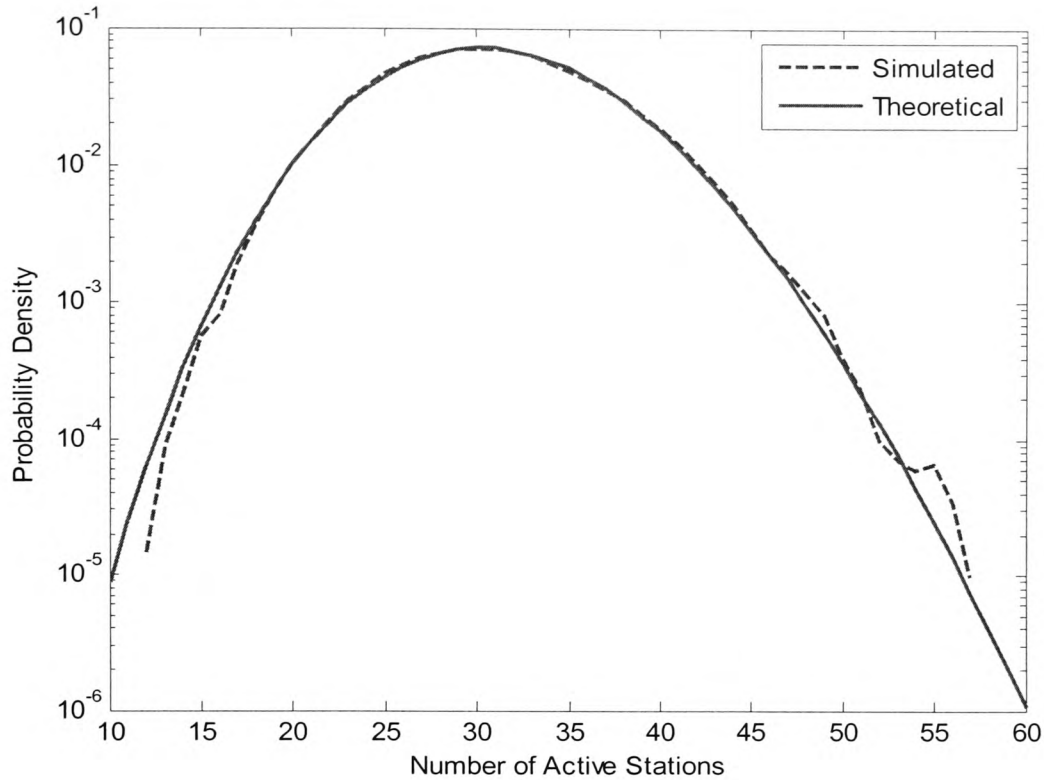


Figure 7.6: PDF of traffic distribution for 620 stations

Each station is allocated its maximum allotted bit-rate during each call, removing the need for extra complexity in terms of modelling traffic types at each station and creating an appropriate algorithm for bandwidth allocation. For a full overview of these satellite networking concerns see [134]. While this extra complexity may better represent a fully-fledged broadband system, it would not add any significant insight into the availability of the service for different FMT scenarios.

7.4 A Dual-channel Protocol for Bi-directional Satellite Internet with FMTs

A dual-channel protocol was envisaged primarily as a method of utilising the higher availability of a lower frequency channel that is part of an FMT scheme that includes frequency diversity as an enabling technology for V band utilisation. The major benefit of separating control and traffic information is the ability to provide different burst rates for the traffic and control data, leading to different availabilities for both channels. This allows the protocol to be tested for scenarios with and without frequency diversity, while still being able to sacrifice bandwidth for availability in the case where frequency diversity is unavailable.

7.4.1 Protocol Structure

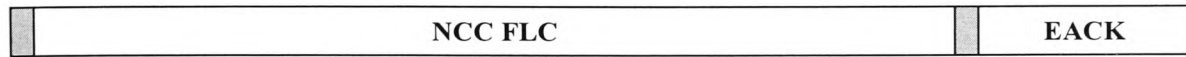
7.4.1.1 Background

Several examples of protocols that are able to pass information for FMT switching already exist [13, 75, 76] as well as the DVB-RCS standard [11]. These can be either centrally controlled such as those in [11, 13], or controlled in a distributed fashion as is the case in [75, 76]. Centrally controlled access schemes tend to be simpler than their distributed counterparts, while sacrificing efficiency in terms of end-to-end delay of network traffic. In addition, recovery from errors is more reliable with centrally controlled access schemes. The distributed access schemes of [75, 76] were a logical progression from the centrally controlled scheme of [13], and a centrally controlled scheme has been opted for in this case as it seems prudent to begin with a protocol that is more stable and less complex. The following protocol was created using the aforementioned protocols as inspiration, the main difference in this case being the separation of the control and traffic channels. Since the protocol is only primarily of

Chapter 7 – Preliminary System Design and Pre-processing

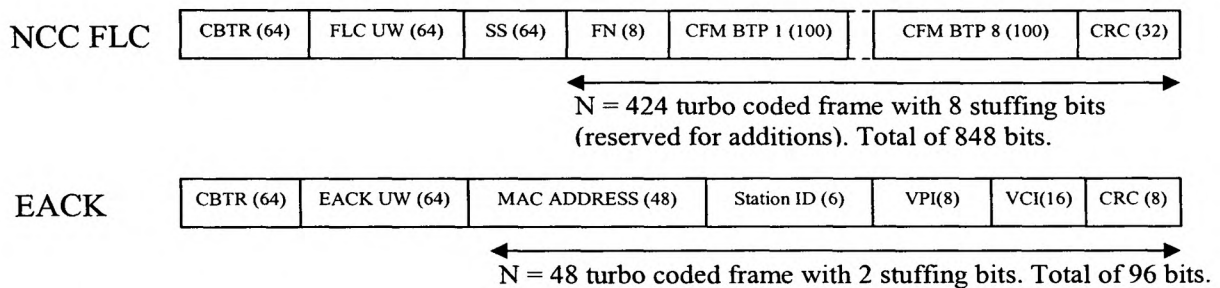
concern for creating link budgets in this case, the exact format of each of the cells has not been developed further as it would serve no purpose.

7.4.1.2 Forward Control Channel (FCC)



NCC FLC Network Control Centre Forward Link Control

EACK Entry Acknowledgement



The acronyms used in the FCC are as follows:

- CBTR Carrier and Bit Timing Recovery – Used to obtain phase lock and fine tune frame timing.
- FLC UW Forward Link Control Unique Word – Identifies the start of a sequence of data.
- SS Signal to Noise Ratio Sequence – A known sequence that is used to estimate the link C/N ratio.
- FN Frame Number – to ensure VSATs burst in the correct frames.
- CFM BTP Capacity and Fade Mitigation Burst Time Plan – A precise definition of the burst time for each station.

Chapter 7 – Preliminary System Design and Pre-processing

- ACK UW Acknowledgement Unique word – Identifies the start of EACK data.
- VPI Virtual Path Identifier used for ATM traffic addressing – Network addressing.
- VCI Virtual Channel Identifier – Network addressing.

The CFM BTP can be further reduced into its component parts as follows:

CFM BTP	Station ID (6)	SFN (8)	NF (8)	IBC (4)	IBS (16)	IBD (16)	OBC (4)	OBS (16)	OBD (16)	R _s (2)	Cr (3)	Mod (1)
---------	----------------	---------	--------	---------	----------	----------	---------	----------	----------	--------------------	--------	---------

the acronyms for which are:

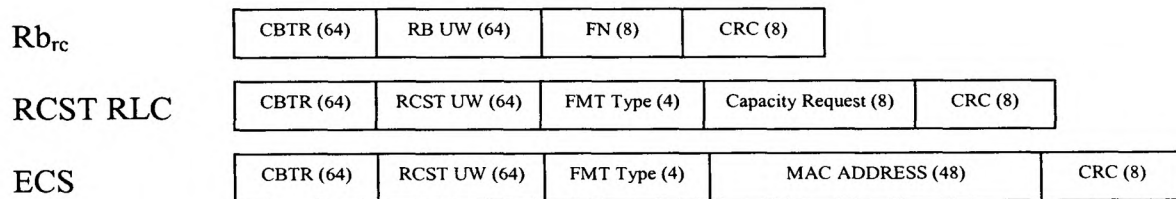
- SFN Start Frame Number
- NF Number of Frames
- IBC Inbound Carrier
- IBS Inbound Start
- IBD Inbound Duration
- OBC Outbound Carrier
- OBS Outbound Start
- OBD Outbound Duration
- R_s Symbol Rate
- Cr Code Rate
- Mod Modulation Type

7.4.1.3 Return Control Channel (RCC)



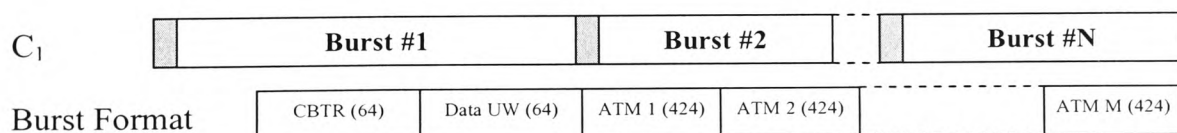
The acronyms used in the RCC are as follows:

- Rb_{rc} Reference Burst Return Control – Phase recovery, timing and frame sequencing.
- RCST RLC Return Channel Satellite Terminal Return Link Control – VSAT FMT resource requests and a placeholder for capacity requests.
- ECS Entry Contention Slot – Used by terminals to log on.



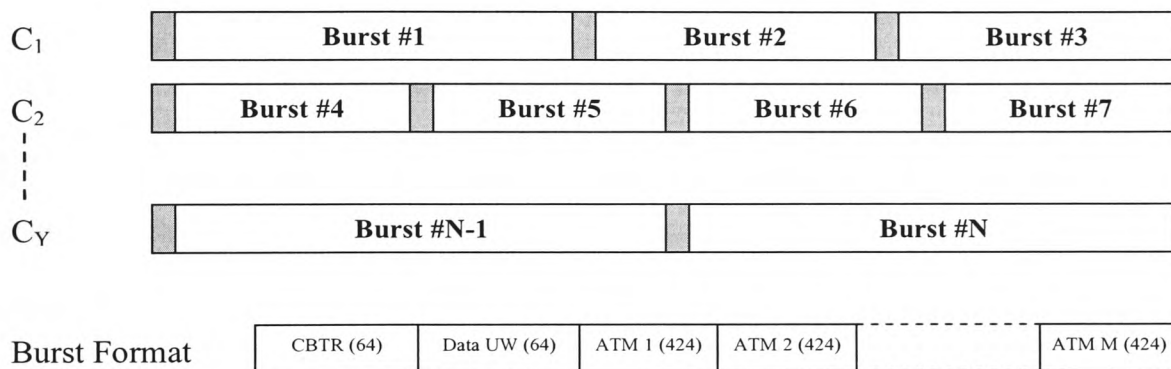
7.4.1.4 Forward Traffic Channel

The forward traffic channel is arranged in a TDMA format with a single carrier as this channel benefits from the high power output and larger antenna of the hub station, allowing a much higher burst rate.



7.4.1.5 Return Traffic Channel

Unlike the forward traffic channel, the return traffic channel is arranged in an MF-TDMA resource sharing scenario, where a subset of the total stations share a carrier between them, but the total number of stations is split over several carriers in order to reduce the required burst rate of the VSAT terminals. The number of bursts on any one carrier is not fixed but variable so that stations can be arranged between the various carriers in an optimum configuration depending on what resources are required due to the fading conditions.



7.4.2 Protocol Functionality

A VSAT terminal that is powered on uses the following procedure to log on to the system. At power on, the VSAT enters a listening state. The carrier is recovered with the CBTR segment that is present on all channels and with this information the VSAT attempts to log-on by sending a burst in the ECS of the RCC. In the event of a collision with another station attempting to access resources, both stations wait a randomly determined number of frames before attempting to retransmit the ECS.

The ECS contains the station's unique MAC address for purposes of verifying whether the station is permitted to use the service and in order to ensure that the terminal knows

which station ID it has been assigned. The ECS also contains the stations fade level, as when the NCC is calculating resource allocation it is important to know how much resource will be required by the station due to fading. This could have an effect on whether the call is admitted or not. The fading level is calculated from the SS that is monitored during the listening state.

If the station's request for capacity is successful, the NCC will respond in the EACK burst with the VSAT's MAC address and an allotted station ID.

Once synchronisation is achieved the VSAT can request capacity for transmitting traffic. The VSAT transmits its fade level and capacity request on its assigned RLC burst, the NCC calculates the full BTP and in the NCC FLC of the next available frame containing the station's ID it transmits, it assigns each station some bursts on the traffic channel based on the current fade level. The VSAT may not transmit on the traffic channel until it receives its BTP. Since only 16 CFM BTPs are transmitted per NCC FLC, it could take up to 4 frames before a BTP is received.

The CFM BTP contains the station ID that the BTP is intended for, the starting frame number for the bursts, the number of frames this BTP is valid for, the return channel carrier, the return channel start time and the return channel duration. However, when there is attenuation at the VSAT side, both forward and return traffic must be protected and therefore the outbound traffic is also sent in bursts to each station instead of as a single TDM stream which would not be able to incorporate variable transmission parameters for each station. Therefore a forward channel carrier, start time and start duration are also specified on the control channel. Finally the CFM BTP contains the

Chapter 7 – Preliminary System Design and Pre-processing

FMT parameters that the station requires to transmit on the traffic channel, a burst symbol rate, a code rate and a modulation scheme.

The VSATs communicate with the NCC via the return control channel, which has been assigned a different carrier to the forward control channel in order to design two heavily protected links that have different EIRPs.

Any station wishing to terminate a connection may do so by use of the capacity request segment of the RCST RLC, and may again request resources using the ECS.

It is assumed that each VSAT can only transmit on a single carrier per frame as is the case for the hardware in [13]. This means that a burst from one station can extend over an entire frame in the worst case scenario. For any change in FMT, the NCC will inform the appropriate VSAT of the change via the CFM BTP and both will be able to switch FMT schemes on an agreed frame.

In order to operate a site diversity scheme, both stations must communicate their own fade levels so the one with the best fading environment can transmit the traffic. However, this can be achieved on the physical link and does not require the resources of the satellite channel.

FEC coding will be as per the DVB-RCS standard. Since this is a turbo code, encoding must be performed on a frame basis (a coding frame, as opposed to an MF-TDMA frame). The control channel is protected with a fixed amount of redundancy although the traffic channel bursts may be protected with a variable coding rate as indicated to

the RCST in the CFM BTP. Each burst type is also protected by an 8 or 32 bit CRC for error detection after FEC decoding.

The CFM BTP of the forward control channel is protected with an 848 bit turbo code and the EACK is protected with a 96 bit turbo code whereas the return control channel is protected with a standard $\frac{1}{2}$ rate convolutional code of constraint length 7 since the size of each element is too small to justify an entire turbo coded frame of 96 bits (the minimum size for the DVB-RCS turbo coding scheme in use). The traffic channels are protected with a 424 bit turbo code for each ATM cell.

7.4.3 Control Channel Availability

An iterative process was used to design a link budget for each link that would balance the links in the forward and return direction for both the traffic channel and the control channel. An overview is provided here for the control channel only, comparing the availabilities of the channel at Ku band and V band. Individual link budgets for the traffic channel are necessary as they depend on the FMT scenario under investigation.

Table 7.3 shows the basic parameters for the overall system which remain constant for each of the link budgets. The majority of these values are typical for VSAT networks, but the satellite EIRP and G/T values were obtained from the Atlantic Bird 1 satellite data sheet [135] and the hub and VSAT antenna sizes were calculated by an iterative process of link budget design. This process was complicated by the fact that the antenna is the same for two different frequencies which directly affects the link budget.

Hub antenna diameter	2 m
VSAT antenna diameter	0.8 m
Hub antenna efficiency	68%
VSAT antenna efficiency	60%
Maximum power available at hub	200 W
Maximum power available at VSAT	2 W
Satellite G/T at Ku band	8 dBi/K
Satellite EIRP at Ku band	51 dBW
Earth segment system noise temperature at Ku band	150 K
Space segment system noise temperature at Ku band	500 K
Earth segment system noise temperature at V band	350 K
Space segment system noise temperature at V band	900 K
Satellite range	38572 km
Transponder bandwidth	36 MHz
Filter roll-off factor	0.35

Table 7.3: Constant parameter values for VSAT network

From these initial parameters, the link budgets for the forward and return channels were designed by the following procedure:

1. Calculate the bit rate, symbol rate and bandwidth for the control channel under consideration using a frame duration of 20 ms, a guard time of 5 μ s between each burst and the number of bits required in the frame.
2. From satellite G/T at Ku band and space segment system noise temperature at Ku band, calculate the satellite antenna gain at Ku band.
3. From Ku band antenna gain and EIRP at Ku band, calculate the available power onboard the satellite.
4. From the satellite antenna gain at Ku band and the ratio of frequencies between V band and Ku band, calculate the satellite antenna gain at V band.
5. Calculate both hub and VSAT earth station antenna gains.
6. Calculate free space path loss.
7. Calculate earth segment and space segment system noise powers.

8. Calculate the required C/N ratio for a BER of 10^{-6} .
9. Calculate the received power at the satellite on the uplink, assuming 0.5% of the available power is used for the forward control channel and 20% of the available power is used for the return channel.
10. Calculate the uplink C/N ratio.
11. Calculate the received power at the earth station, assuming that the satellite power used is proportional to the occupied bandwidth as a percentage of transponder bandwidth.
12. Calculate the downlink C/N ratio.
13. Calculate the overall C/N ratio from the uplink and downlink C/N ratios.
14. Calculate the link margins, taking into account the increased noise power on the downlink during rain.

Assuming that bandwidth is cheap in order to find the link margins for a best case scenario for a V band control channel, a code rate of 1/3 is used for both the turbo code on the forward control channel and the convolutional code on the return control channel. This yields required E_b/N_0 values of 2.5 dB (see figure 4.8) and 4.2 dB [136] respectively.

Using these parameters, for the forward control channel, an uplink margin of 26.7 dB and an effective downlink margin of 12.6 dB are obtained at V band. For the return channel, an uplink margin of 11.1 dB and an effective downlink margin of 22.2 dB are obtained at V band. The lowest of these values are the ones associated with the VSAT side of the link and these can be used to estimate an annual availability [77]. An effective downlink margin of 12.6 dB at 40 GHz yields an availability of approximately

99.9% and an uplink margin of 11.1 dB at 50 GHz yields an availability of approximately 99.3%, considering attenuation due to rain for a British climate.

In contrast to this, if the control channels are located at *Ku* band, and a code rate of $\frac{1}{2}$ is used for both the turbo code on the forward control channel and the convolutional code on the return control channel, the required E_b/N_0 values become 2.8 dB (see figure 4.8) and 4.5 dB [136] respectively.

The forward control channel at *Ku* band now has an uplink margin of 21.1 dB and an effective downlink margin of 5.5 dB, while the return control channel has an uplink margin of 6.1 dB and an effective downlink margin of 9.6 dB. Again the lowest of these values are associated with the VSAT side of each link and were used to estimate the annual availability. Although the margins for attenuation are lower, rain attenuation at *Ku* band is also lower than at *V* band. An effective downlink margin of 5.5 dB at 11.35 GHz gives an availability of approximately 99.999% and an uplink margin of 6.1 dB at 13.87 GHz gives an availability of approximately 99.997% using the ITU-R model [9] since measurements at these exact frequencies are not available.

The Matlab code used to calculate these link budgets can be seen in appendix Q.

7.5 *Conclusions*

The control channel at Ku band, despite having a less costly coding scheme offers a far better availability than if it were located at V band. This alone makes a very strong case for the use of frequency diversity as control channel errors affect the operation of the entire system, not just individual stations. Nevertheless, the traffic channel for several schemes will be investigated in the following chapter, one of which does not include frequency diversity.

The simulations in the following two chapters make use of several elements described here. Firstly, the bit rates and consequently symbol rates and bandwidths of all links were calculated directly from the protocol structure, taking into account a frame duration of 20 ms and guard times of 5 μ s between bursts. Secondly, the dataset used for the final simulations was created by the pre-processing described in this chapter. Finally the traffic scenario in chapter 9 was based on the traffic assumptions outlined in this chapter.

Chapter 8

Preliminary Investigation of Combined FMT Performance

8.1 Introduction

Before deciding upon a final design to simulate in full, three different case scenarios were investigated in order to gather statistics that could highlight the advantages and disadvantages of site diversity, frequency diversity and combined adaptive signal processing techniques. This analysis would also provide information that could feed back into to a finalized design of the link-layer protocol and various link-budget considerations. These simulations were run with a minimum-complexity approach that avoided coding of the underlying physical-layer elements of the system. Instead, only the link-budgets and 1Hz sampled time series were used to generate statistics of the simplified scenarios of each FMT.

8.2 Design Considerations

The following assumptions apply to the results obtained herein:

1. Site diversity switching is instantaneous. That is, at any sample, of the two diversity stations the one with the lowest attenuation is chosen with no delay in switching from one to the other.
2. The VSAT's ability to measure the overall link C/N ratio is perfect. The time series of attenuation acquired by beacon measurement at RAL is used directly in the calculation of uplink C/N, downlink C/N and overall C/N, with no errors introduced from the measurement of C/N using the Signal to Noise Sequence (SS) of the control channel [52].

3. The duo-binary turbo coding scheme employed [11] performs as the measurements suggest [137] at an error rate of 10^{-6} . The coding scheme itself is not applied here, only the assumption that it performs as it should at the specified BER and for the specified code rate and block size.
4. The resource management algorithm at the hub is able to utilize the full total reserve capacity of the system regardless of whether the real system would be able to segregate the traffic in such a way. If the total frame duration of 0.4 seconds over the 20 carriers of the return link is able to accommodate the total duration required by all the stations that are not in outage, it is assumed that the capacity is sufficient. No consideration is taken for how the stations will be arranged over the 20 carriers.
5. The satellite transponder can operate at the power output specified for the Atlantic Bird 1 satellite with no back-off for both forward and return channels. No interference due to transponder non-linearity is taken into account in the C/N calculation and no output back-off is assumed.
6. The control channels function without error. This allows the traffic channel to be analysed without considering the performance drop that would occur in the event of control channel errors.
7. No uplink power control is available.
8. The assumed traffic parameters described in chapter 7 are ignored in favour of a system where all channels are utilized constantly by 58 VSATs. This would give a very pessimistic view if the results were to be taken as indicative of performance, but at this stage the relative performance of the different FMTs is what is of interest. Simulation of traffic is a significant burden in terms of computation time which is the principal reason for its omission at this stage.

Chapter 8 – Preliminary Investigation of Combined FMT Performance

These assumptions lead to a system simulation that is able to compare different FMTs. However, in reality assumptions 1 to 6 would add varying degrees of performance degradation.

Site diversity switching in reality would involve some delay due to buffering and comparison of received frames, and may also include hysteresis if switching adds a variable delay to the system, leading to an optimum switching time in the order of several seconds [138].

The way in which the overall C/N is calculated is also prone to inaccuracy [52] and system outage and resource exhaustion statistics could be adversely affected as a result.

The performance of turbo coding may have some minor impact on the statistics but of all the assumptions made, it will likely have the least impact as the performance should average to the values found in chapter 4 over a large number of samples.

In reality the resource management algorithm may not always be able to fully utilize the available frame duration. The stations may be spread over all the return link carriers with every carrier having a small unused duration that is unusable because each station is limited to communicating on a single carrier at a time. For this reason, even if the total frame duration required by all the stations in the system may be slightly lower than the total frame duration available on all the carriers, resource exhaustion may still occur due to the individual burst times not dividing evenly between the carriers.

Chapter 8 – Preliminary Investigation of Combined FMT Performance

The interference of non-linear transponders, or more accurately, the back-off that is applied to the amplifier on board the satellite to reduce the interference could also be a factor in the overall C/N ratios. However, this only seriously affects the return link which uses several carriers, and the downlink C/N on this link is very high in comparison with the uplink, so the overall C/N should not be too far from reality.

Control channel errors would cause significant performance degradation since in this event all stations could be affected regardless of their own current meteorological conditions.

The lack of power control will result in poor performance on the forward link, but this can later be adjusted when it is included for the final system simulation in the following chapter.

Assumptions 7 and 8 are pessimistic rather than optimistic, but the overall objective must remain as one of relative performance rather than absolute performance since the exact effect of the combination of the above assumptions is not known.

Despite these assumptions, it is useful to obtain statistics for this scenario which benefits from a significantly lower coding and execution time than a full system simulation would entail. The results from this preliminary stage were obtained several times in a recursive process, from which a final protocol and link budget were obtained (see chapter 7 for the protocol used) for a more detailed, but more time consuming final simulation that appears in the following chapter.

Chapter 8 – Preliminary Investigation of Combined FMT Performance

In each case scenario three information files were loaded for use in the simulation. The files contained the dates of the days that were chosen in the pre-processing stage (chapter 7) and the forward and return link budget information that was calculated separately for each scenario. Although the link budgets were calculated separately, the percentage of resources allocated for FMT utilization and the total bandwidth for each scenario was kept constant to facilitate a useful comparison.

The method used in each case was to take consecutive days from the dataset for each station including the hub. This led to a matrix of dimensions 86400 rows by 59 columns. It is assumed that in a real system, stations would be separated by a large distance and therefore would be highly uncorrelated in terms of weather effects. This assumption is justified since satellite broadband is likely only to be used by those living in remote regions and not in major population centres where terrestrial broadband services would be the preferred option. It is also assumed that rain events occurring at the same time in consecutive days are unlikely and therefore that measurements that are in reality at a fixed spot on the Earth but separated by twenty-four hours can be assumed to be statistically uncorrelated in a similar way as measurements taken simultaneously at large separation distances.

Since the resource provided is assumed to be constantly occupied by 58 stations, a certain portion of it must be allocated exclusively for FMT purposes in this simulation. For the purpose of creating a link budget, the FMT capacity was chosen to be 25% of total capacity or 33.3% of nominal capacity, i.e. 5 ms of the frame duration on the forward link, or 5 spare carriers on the return link. The former notation of FMT capacity in relation to the total capacity will be used in future references to FMT capacity.

Chapter 8 – Preliminary Investigation of Combined FMT Performance

However, statistics of the required frame duration to accommodate all stations were gathered to indicate what percentage of FMT resources would be required for the system to operate at various availabilities.

In order to compare the performance of each simulation, a term must be carefully selected to represent the case where the current resource requests demand more time than is available in the frame, or group of frames in the case of the return link. If traffic alone were the cause of certain stations being unable to access the shared resource then blocking probability would be an acceptable term but since both the number of stations and their respective burst lengths are now factors another term is useful to avoid confusion. Outage probability is already defined as the likelihood of the C/N ratio dropping below the dynamic range of the FMT and is therefore also unsuitable. Frame exhaustion probability has been chosen to represent this factor for the case at V band with a signal processing FMT and with a hybrid FMT, whereas Ku band exhaustion is used for the case with only frequency diversity. The term resource exhaustion will be used to refer to this parameter in a general sense.

Unlike blocking, frame exhaustion can lead to two different scenarios. Stations requesting resources from the hub may be denied as with blocking but stations that are currently transmitting may also be denied resources to overcome rain-induced fading. The hub may manage this situation in different ways but here it is just assumed that the station in the highest fade category would be dropped first if an FMT resource request were to cause frame exhaustion as this would be the station most likely to go into outage and would also free the largest amount of resources. In the case where a station

attempts to log in to the system but not enough resources are available, the station can be told to retry after a random amount of time has expired, as described in chapter 7.

8.3 *Signal Processing Scenario*

The signal processing scenario made use of adaptive modulation and adaptive coding in combination with an adaptive MF-TDMA scheme to progressively trade burst duration for transmission quality. The nominal modulation scheme was QPSK with BPSK providing a level of protection and the nominal coding scheme was a 6/7 code-rate with various code-rate increments providing extra protection. The overall scheme was as follows:

Modulation	Code Rate	C/N Required (dB)	Forward/Return Link Burst Durations (ms)
QPSK	6/7	8.5	0.254 / 5.3
QPSK	4/5	7.9	0.271 / 5.7
QPSK	3/4	7.2	0.289 / 6.0
QPSK	2/3	6.7	0.325 / 6.7
QPSK	1/2	5.8	0.432 / 8.9
BPSK	4/5	4.9	0.542 / 11.3
BPSK	3/4	4.2	0.578 / 12.0
BPSK	2/3	3.7	0.649 / 13.5
BPSK	1/2	2.8	0.864 / 17.7

Table 8.1: FMT adaptive modulation and coding scheme

Two versions of this scheme were investigated, one that used site diversity at the hub which was created using the method discussed in chapter 7, and another without site diversity at the hub. Where site diversity was used, each sample of the data was compared with a sample from the same series, separated in time by 870 samples, or 14.5

minutes which is the separation time determined in chapter 5. The sample with the lower attenuation was used in a new time series.

The parameters most relevant to the link budget for the forward link can be seen in table 8.2 below. These assume a satellite with parameters equivalent to those of Eutelsat's Atlantic Bird 1 satellite [135]. It is assumed here that the hub is located within the second inner-most contour of the footprint representing a 1 dB loss from the largest G/T and covering the majority of Europe. For the sake of simplicity it is assumed that all the VSATs are located at a position with a 3 dB loss from the inner-most contour of satellite EIRP, which would provide similar coverage of Europe as the hub location footprint. The satellite G/T and EIRP are reformulated for a V band scenario since the gain and system noise temperature are higher at these frequencies. This is achieved, as is the overall link budget, by the same process as described for the control channel link budgets in chapter 7, with the exception that the bit rates, symbol rates and bandwidth are now calculated according to the amount of traffic, and not the protocol structure.

Hub antenna diameter	2 m
Hub antenna efficiency	68%
VSAT antenna diameter	0.8 m
VSAT antenna efficiency	60%
Hub power	5 W
Satellite G/T at Ku band T_s of 500K	8 dBi/K
Satellite G/T at V band T_s of 900K	16.7 dBi/K
Earth station system noise temperature	350 K
Satellite EIRP at Ku band	51 dBW
Satellite EIRP at V band	61.8 dBW
Number of carriers	1
Burst rate of each carrier (QPSK modulation, $5\mu s$ guard time)	53.31 Mbps
Occupied transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	35.98 MHz
Minimum C/N for BER of 10^{-6} (nominal FMT setting)	8.51 dB
Total information bit rate available for 58 stations at 512 kbps	29.7 Mbps
Total information bit rate including ATM header overhead	33.2 Mbps
Uplink frequency	50.46 GHz
Downlink frequency	39.56 GHz
Miscellaneous losses	0.5 dB
Gaseous losses at uplink frequency	3 dB
Gaseous losses at downlink frequency	1 dB

Table 8.2: Most significant link budget parameters for the signal processing forward link

The parameters most relevant to the return link are shown in table 8.3 below. The placement of hub and VSATs is assumed to be the same as the forward link and losses are determined accordingly.

Hub antenna diameter	2 m
Hub antenna efficiency	68%
VSAT antenna diameter	0.8 m
VSAT antenna efficiency	60%
VSAT power	0.8 W
Satellite G/T at Ku band T_s of 500K	8 dBi/K
Satellite G/T at V band T_s of 900K	16.7 dBi/K
Earth station system noise temperature	350 K
Satellite EIRP at Ku band	51 dBW
Satellite EIRP at V band	61.8 dBW
Number of carriers	20
Burst rate of each carrier (QPSK modulation, $5\mu s$ guard time)	397.2 kbps
Occupied transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	6.16 MHz
Minimum C/N for BER of 10^{-6} (nominal FMT setting)	8.51 dB
Total information bit rate available for 58 stations at 64 kbps	3.71 Mbps
Total information bit rate including ATM header overhead	4.92 Mbps
Uplink frequency	50.42 GHz
Downlink frequency	39.52 GHz
Miscellaneous losses	0.5 dB
Gaseous losses at uplink frequency	3 dB
Gaseous losses at downlink frequency	1 dB

Table 8.3: Most significant link budget parameters for the signal processing return link

A flow diagram for the analysis is shown in figure 8.1. The Matlab code for this analysis is provided in appendix R.

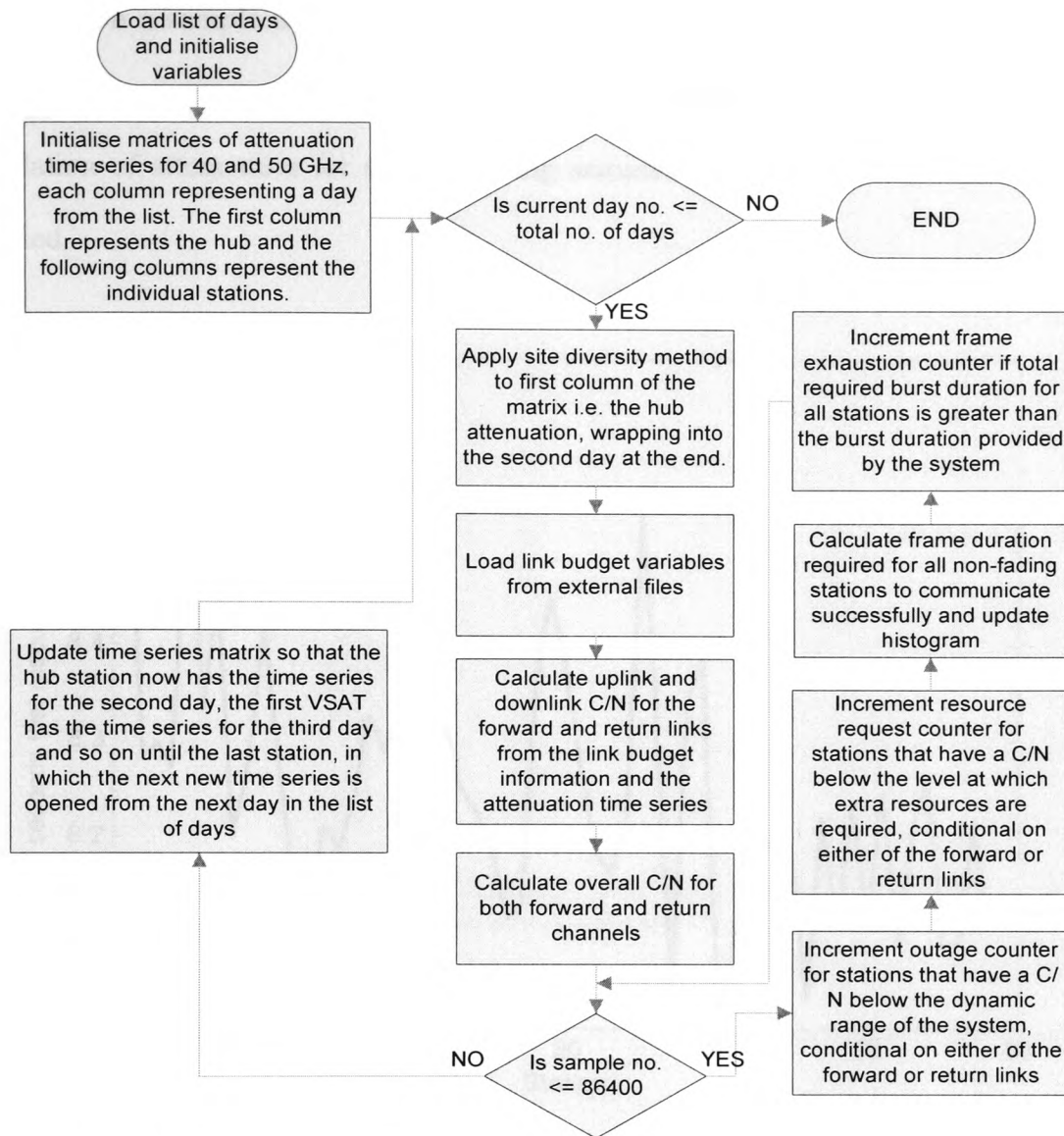


Figure 8.1: Flow diagram of Matlab program of Signal Processing FMT with Site Diversity

The signal processing scenario has 137568 samples in which the capacity requests of all stations cannot be accommodated by the hub, i.e. a frame exhaustion probability of 0.825%.

Figure 8.2 shows the number of FMT resource requests made by each station for the entire duration, where a resource request was defined as being made when the overall C/N for a station was below the threshold where the FMT was required. This analysis

Chapter 8 – Preliminary Investigation of Combined FMT Performance

was done on a sample by sample basis, assuming no delay. The results show that the variance of resource requests per station is low and therefore the assumption of low correlation of attenuation for corresponding samples between one day and the next is justified.

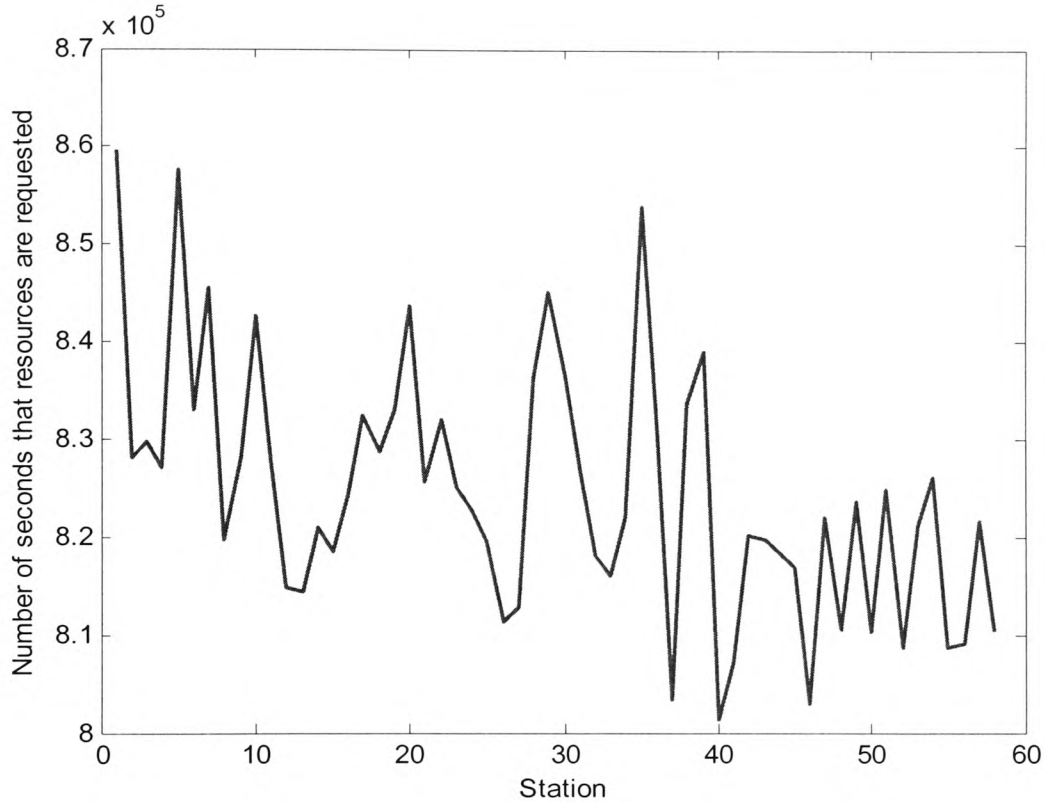


Figure 8.2: Number of resource requests per station for signal processing scheme with site diversity

Figure 8.3 shows the number of outages per station, where an outage is defined as a C/N below which the FMT can no longer compensate, i.e. the dynamic range of the FMT has been exceeded. This result also justifies the assumption of low correlation of corresponding samples between days since the variance is low.

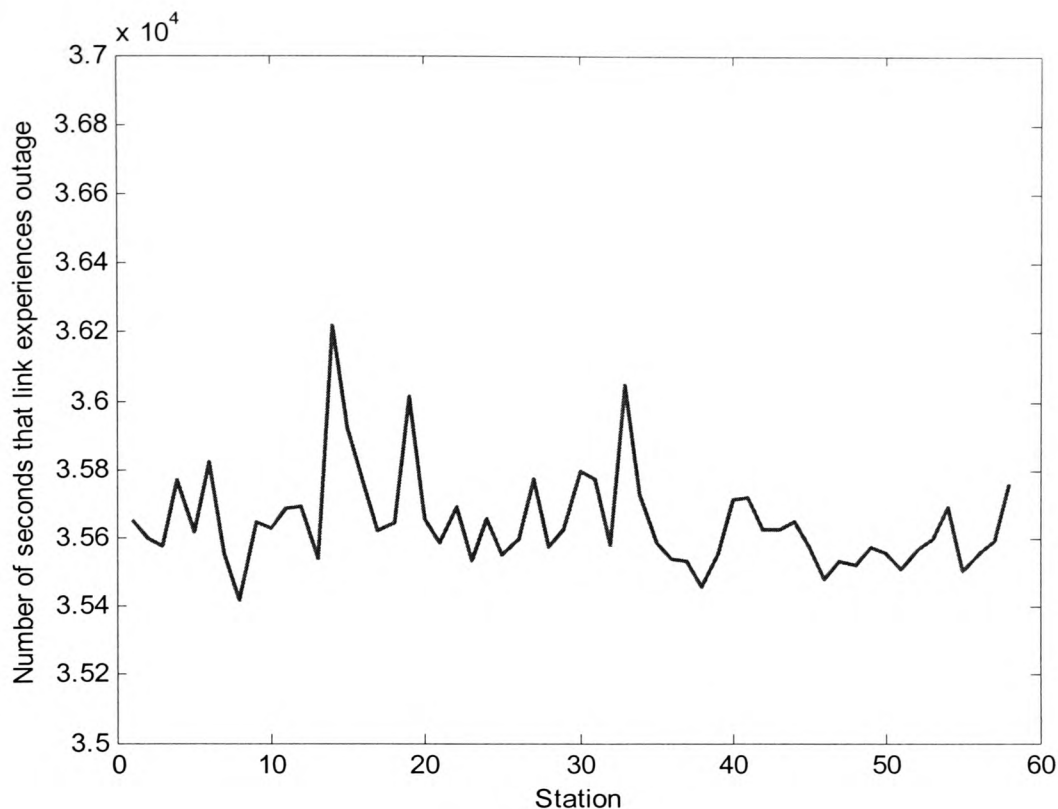


Figure 8.3: Number of seconds in outage per station for signal processing scheme with site diversity

Figure 8.4 shows the probability density of the frame duration required for all stations not in outage to transmit with the necessary FMT parameters for the forward link. The graph clearly shows that the vast majority of durations are at 75% of capacity, i.e. nominal conditions. There are several occasions in which less than 75% of capacity is used. This can be attributed to hub-side fading which causes outage for many of the links. Similarly, there are instances where there is frame exhaustion and the system capacity required is in excess of 100%. This can also be mostly attributed to lighter hub fading where all stations may be within the dynamic range of the system but would require longer bursts to transmit at the desired BER.

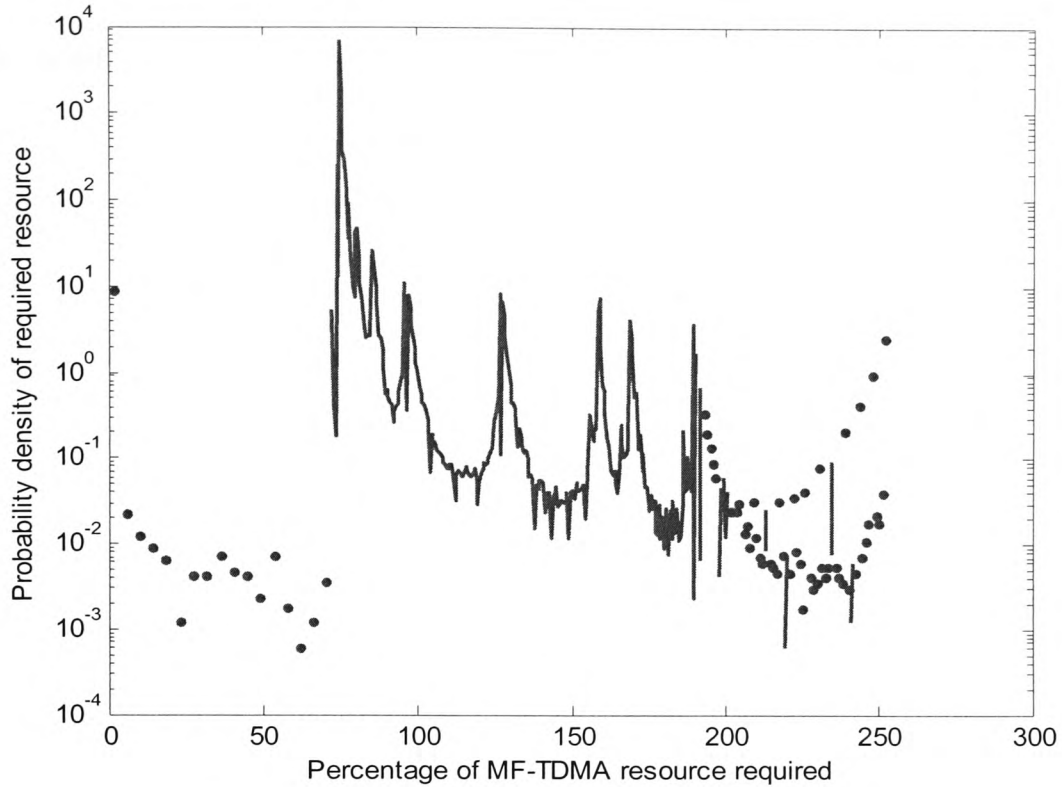


Figure 8.4: PDF of percentage of MF-TDMA frame duration required for all stations to transmit at the chosen BER on the forward link with site diversity

Figure 8.5 shows the cumulative distribution of the frame duration required for all stations not in outage to transmit with the necessary FMT specifications for the forward link. This shows that the probability of frame exhaustion drops rapidly over a small reserve duration and then quickly levels out, requiring a large reserve duration for any significant change in probability of frame exhaustion. The flattening occurs at about 85% of the resources and at a frame exhaustion probability of about 2%. Most efficient use of spare resources occurs at this point, any further resources providing little extra protection to the link in statistical terms. This observation can be attributed to the dominance of the hub-side fading scenario. If there is a high attenuation at both hub diversity sites then every link is affected and will require resources. The protocol and link budget were not designed so that the signal processing FMT could alleviate fades at

the hub, although these fades will be overcome to an extent when uplink power control is included.

From this it becomes apparent that the optimum shared resource to protect against rain-induced fades should be roughly 10% of the total MF-TDMA frame duration. Rather than alter the frame duration of the scheme, future investigation will consider the shared FMT resource to be 10% of the frame duration, or 2ms of a 20ms frame.

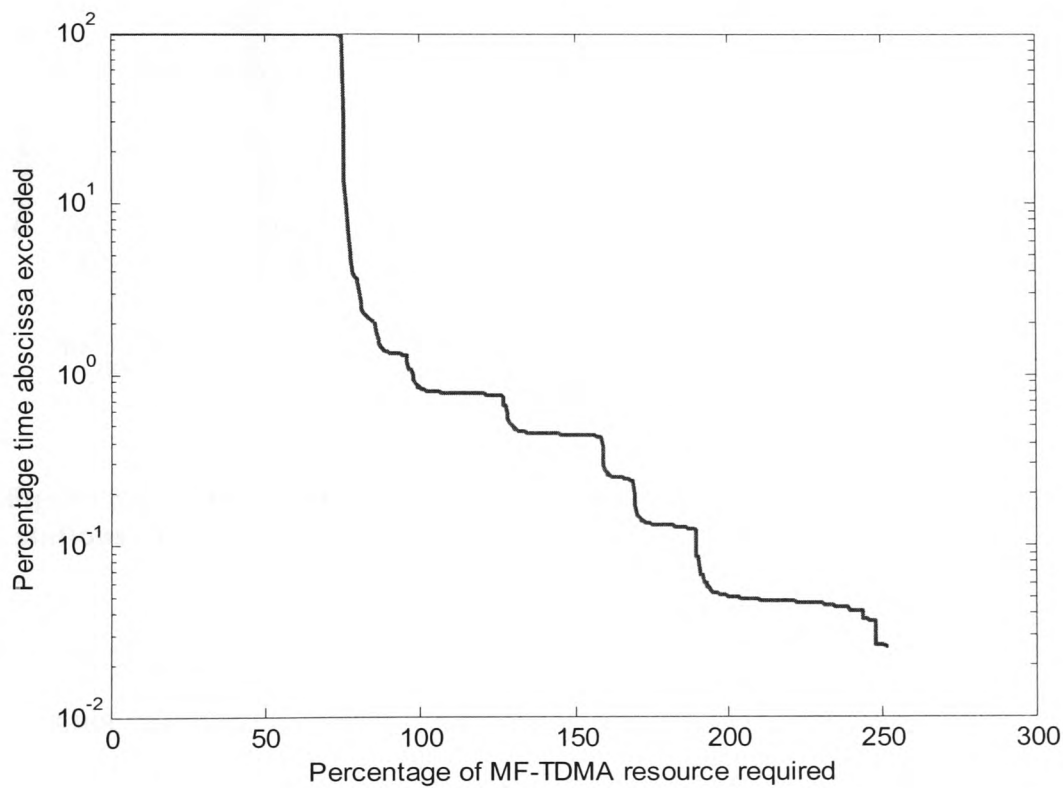


Figure 8.5: CDF of percentage of MF-TDMA frame required for all stations to transmit at the desired BER on the forward link with site diversity

Figure 8.6 shows the probability density of the frame duration required for all stations not in outage to transmit with the necessary FMT specifications for the return link. The frame duration of the scheme is simply the sum of the duration of each carrier, i.e. 0.4 seconds. As stated earlier, no effort has been made to create a detailed burst time plan,

so the burst durations are considered directly against the total duration, with the assumption that no frame duration is wasted.

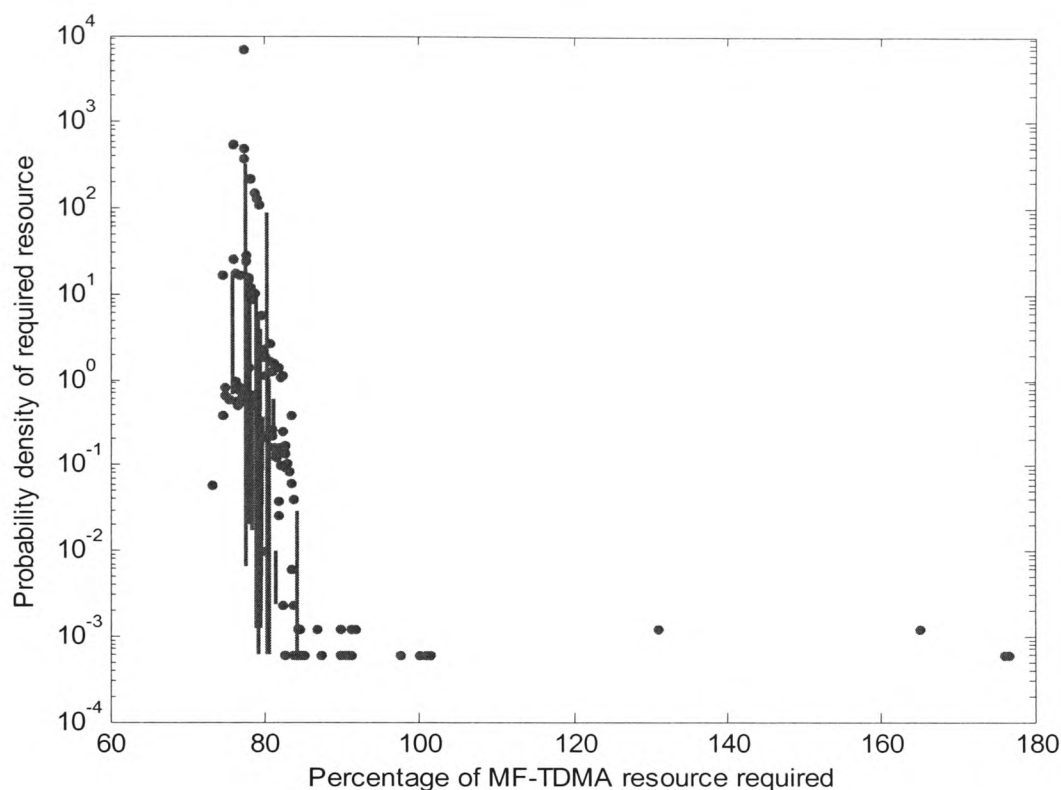


Figure 8.6: PDF of percentage of MF-TDMA frame duration required for all stations to transmit at the chosen BER on the return link with site diversity

Figure 8.7 shows the cumulative distribution of the frame duration required for all stations not in outage to transmit with the necessary FMT specifications for the return link. A similar trend is shown for the return link as the forward link, with the probability of frame exhaustion dropping rapidly as the total MF-TDMA frame duration is first increased above the nominal value of 75% but flattening at about 82%. This occurs at a probability of resource insufficiency of about 0.01%.

From this information it is possible to calculate that the optimum shared resource should be 7% of the total frame duration. One possible reason that the optimum resource is

lower for the return link than the forward link is that when designing the links, the 15 nominal carriers are in fact able to accommodate 60 stations rather than 58, making the total spare capacity equal to 27.5% rather than 25%. When this is considered, the optimum shared resource would be 9.5%, which is closer to the forward link result. The return link cannot be designed with the same resolution as the forward link as extra resources are available as spare carriers, not as extra duration in a single frame. However, the results show that 15 nominal carriers with 2 spare carriers matches the statistics quite well.

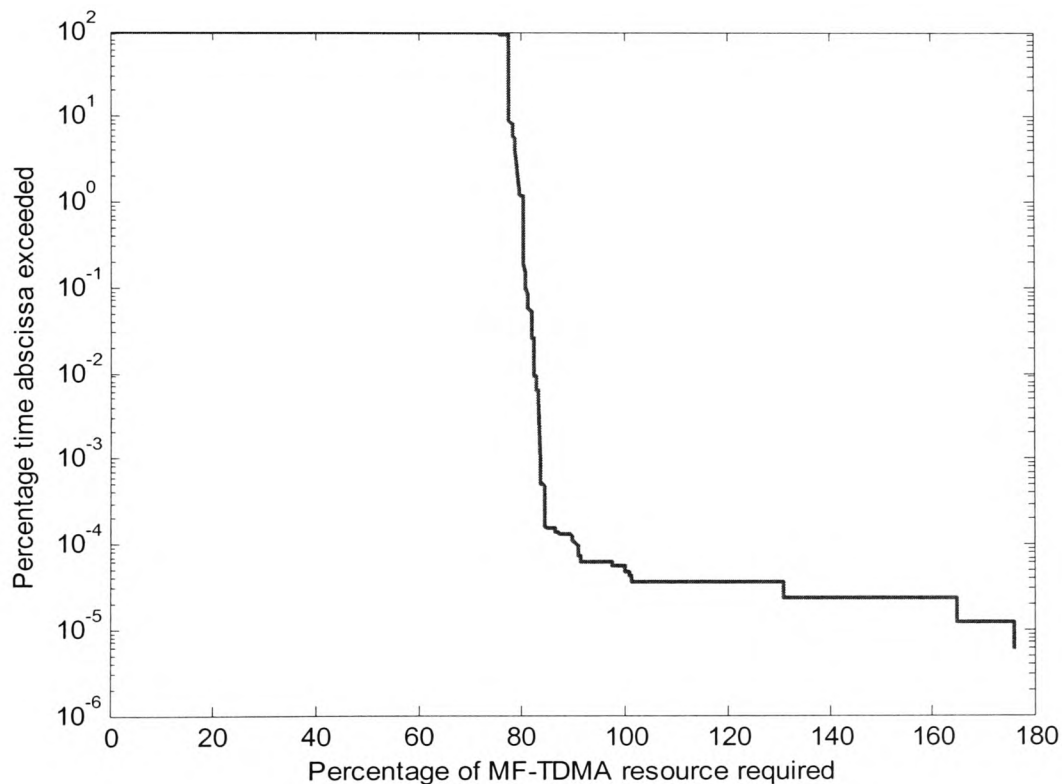


Figure 8.7: CDF of percentage of MF-TDMA frame required for all stations to transmit at the desired BER on the return link with site diversity

The same statistics were generated for a scenario without site diversity at the hub which would be expected to fare worse than a scheme with site diversity. Figures 8.8 to 8.11 show the same parameters as already seen in figures 8.2, 8.3, 8.5 and 8.7 but without

site diversity. The same software approach was taken as for the case with site diversity but with the section of code that implemented site diversity removed. The flow diagram is therefore the same as that presented in figure 8.1 with the exception of the first block in the loop.

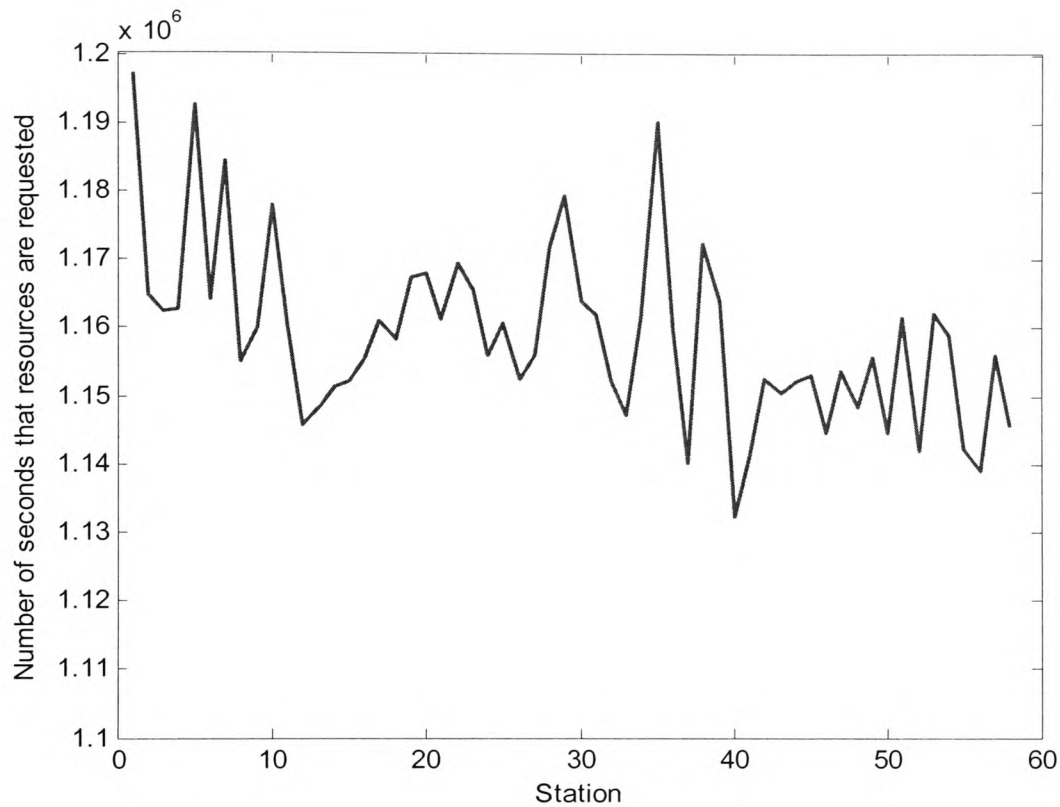


Figure 8.8: Number of resource requests per station for signal processing scheme without site diversity

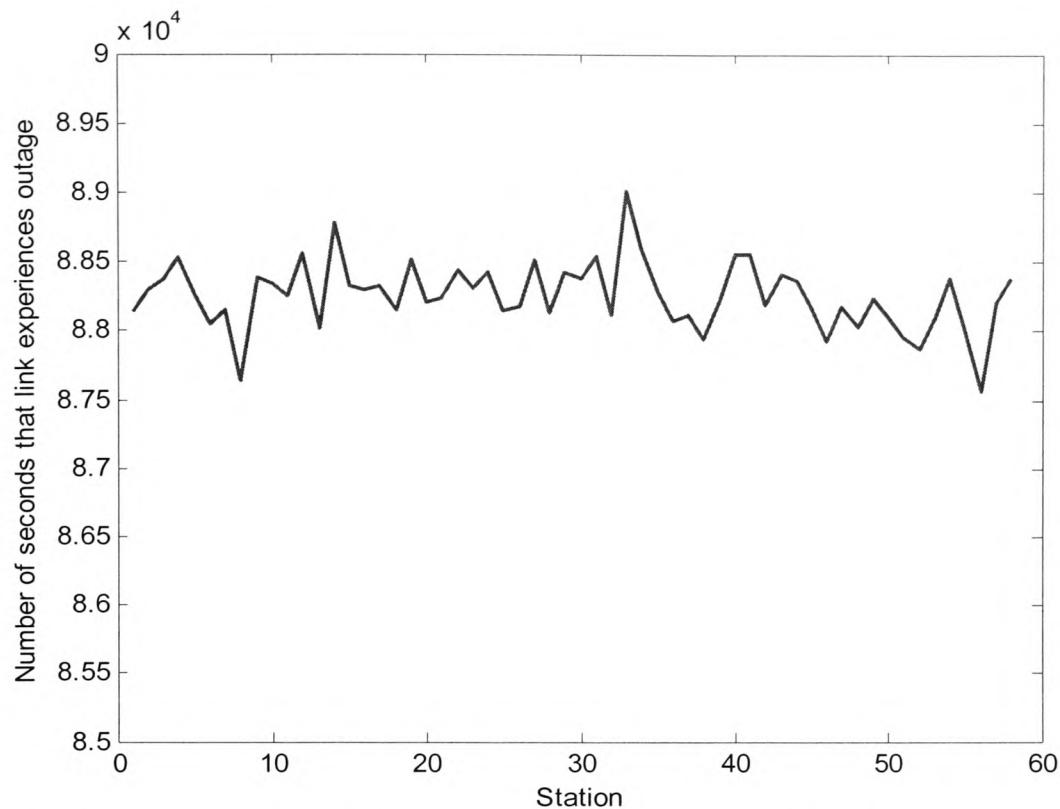


Figure 8.9: Number of seconds in outage per station for signal processing scheme without site diversity

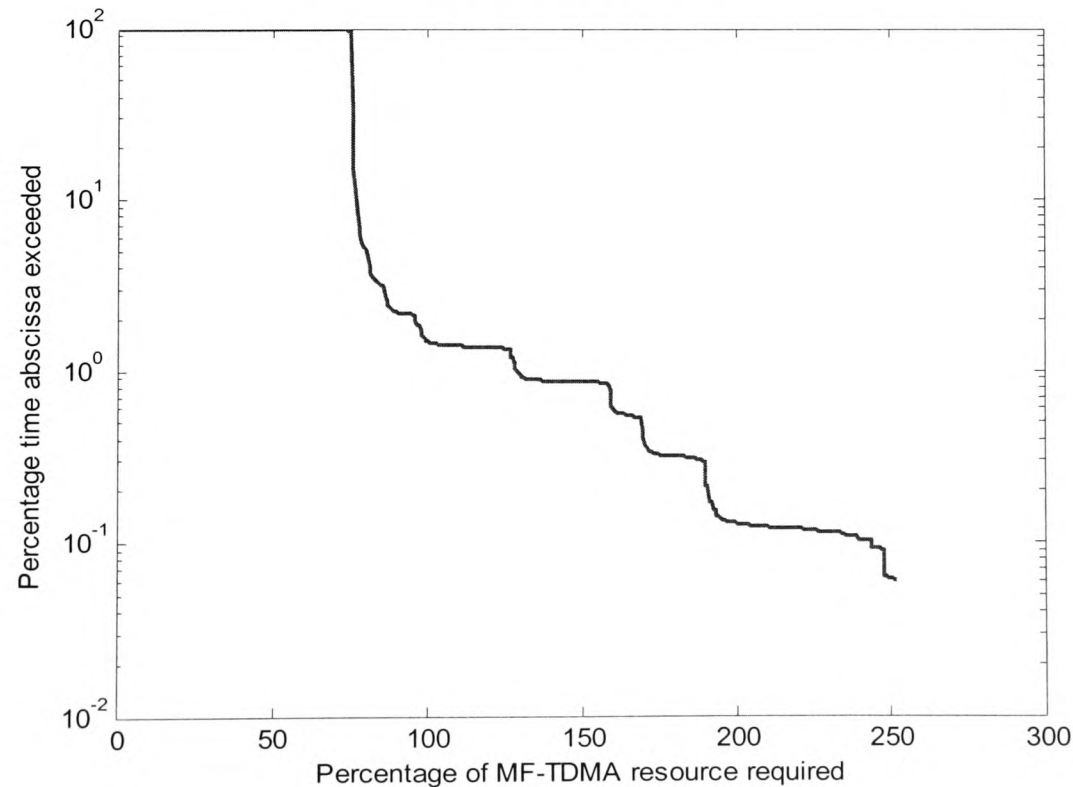


Figure 8.10: CDF of percentage of MF-TDMA frame required for all stations to transmit at the desired BER on the forward link without site diversity

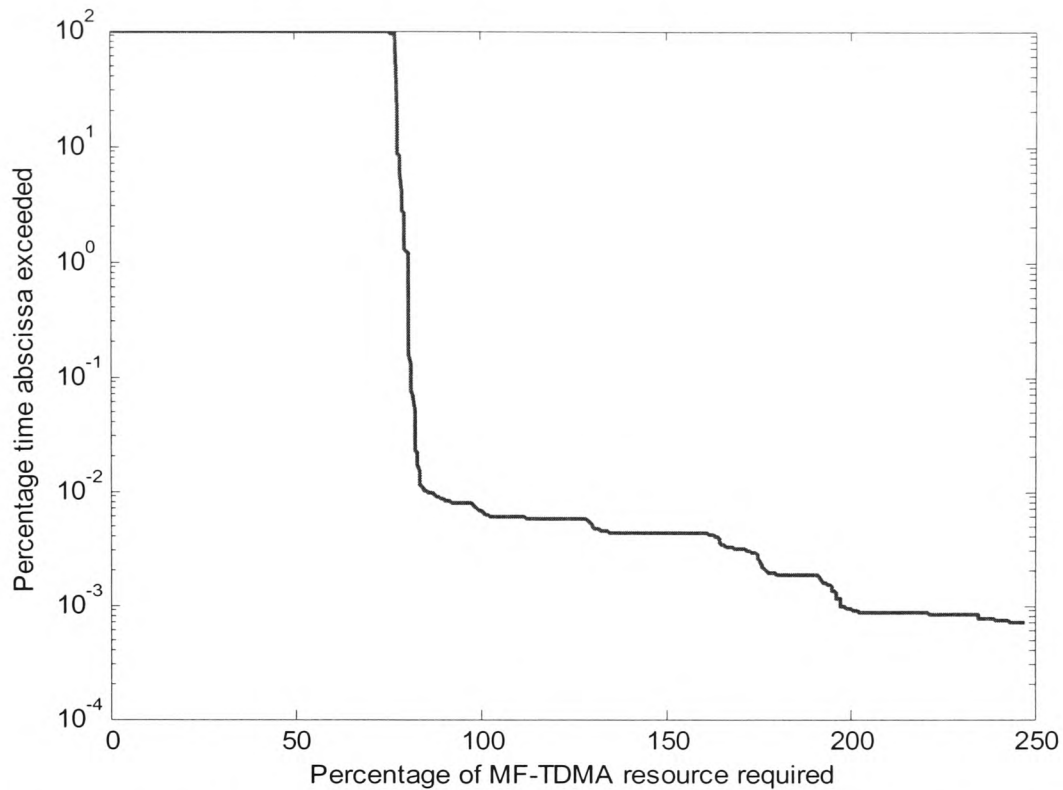


Figure 8.11: CDF of percentage of MF-TDMA frame required for all stations to transmit at the desired BER on the return link without site diversity

This scenario results in 243821 sample instances where all the capacity requests cannot be accommodated by the hub. This is a frame exhaustion probability of 1.46%. This is almost double the number for the scenario with site diversity, a significant difference.

It is clear from these results that the system does not perform as well as the one that uses site diversity. Figure 8.8 shows that resource requests total an average of 322 hours per station for the system without diversity compared with 229 hours with diversity. This equates to a 7% and 5% annual usage of FMT resources per station respectively. In the final design, a lower resource usage dedicated to FMT resources is desirable due to the shared nature of the medium. Resources that are being used for FMT purposes are not available for extra traffic and therefore the blocking probability would increase if site diversity were not used. Figure 8.9 shows that the average annual outage period is

24½ hours without diversity but only 10 hours with. This equates to annual availabilities of 99.47% without and 99.79% with diversity. Figures 8.10 and 8.11 also show that the probabilities of frame exhaustion at the optimum frame duration for the system without diversity are roughly 3% on the forward link and 0.01% on the return link, compared with 2% on the forward link and 0.002% on the return link with diversity.

This gives a clear indication that site diversity is a necessity at the hub as it makes a large impact on both the annual availability of stations due to rain while also reducing the percentage of resources that must be allocated for FMT use, which in turn will positively affect the blocking probability.

8.4 Frequency Diversity Scenario

The frequency diversity scenario assumed that 25% of the total bandwidth was located at *Ku* band, in a channel designed with a fixed margin. This complicated the design slightly for the forward link, as two carriers were now required, one at *V* band holding 75% of the traffic and one at *Ku* band holding the remaining 25%. The return link simply assumed that 3 of 15 carriers were located at *Ku* band. No signal processing was used so the stations were either transmitting at a fixed burst duration in *V* band or at a fixed burst duration in *Ku* band. The *Ku* band channels were assumed to have uplink carriers at 14 GHz and downlink carriers at 12 GHz. Time series at these frequencies were created from the 40 GHz series using several scaling models as described in chapter 7.

Chapter 8 – Preliminary Investigation of Combined FMT Performance

The parameters most relevant to the link budget for the forward link appear in table 8.4 below. These assume the same satellite is used and same footprint losses are applicable as in the signal processing scenario.

Hub antenna diameter	2 m
Hub antenna efficiency	68%
VSAT antenna diameter	0.8 m
VSAT antenna efficiency	60%
Hub power	5 W
Satellite G/T at <i>Ku</i> band	8 dBi/K
Satellite G/T at <i>V</i> band T_s of 900K	16.7 dBi/K
Satellite system noise temperature at <i>Ku</i> band	500 K
Earth station system noise temperature at <i>Ku</i> band	150 K
Satellite system noise temperature at <i>V</i> band	900 K
Earth station system noise temperature at <i>V</i> band	350 K
Satellite EIRP at <i>Ku</i> band	51 dBW
Satellite EIRP at <i>V</i> band	61.8 dBW
Number of carriers at <i>Ku</i> band	1
Number of carriers at <i>V</i> band	1
Burst rate of each <i>Ku</i> band carrier (QPSK modulation, 1/2 rate coding, 5 μ s guard time)	13.78 Mbps
Burst rate of each <i>V</i> band carrier (QPSK modulation, 6/7 rate coding, 5 μ s guard time)	31.16 Mbps
Occupied <i>Ku</i> band transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	9.301 MHz
Occupied <i>V</i> band transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	21.03 MHz
Minimum C/N for BER of 10^{-6} at <i>Ku</i> band	5.51 dB
Minimum C/N for BER of 10^{-6} at <i>V</i> band	8.51 dB
Total information bit rate available for 58 stations at 512 kbps	29.7 Mbps
Total information bit rate including ATM header overhead	33.2 Mbps
<i>Ku</i> band uplink frequency	13.87 GHz
<i>Ku</i> band downlink frequency	11.35 GHz
<i>V</i> band uplink frequency	50.42 GHz
<i>V</i> band downlink frequency	39.52 GHz
Miscellaneous losses	0.5 dB
Gaseous losses at <i>Ku</i> band uplink frequency	0.7 dB
Gaseous losses at <i>Ku</i> band downlink frequency	0.5 dB
Gaseous losses at <i>V</i> band uplink frequency	3 dB
Gaseous losses at <i>V</i> band downlink frequency	1 dB

Table 8.4: Most significant link budget parameters for the frequency diversity forward link

Chapter 8 – Preliminary Investigation of Combined FMT Performance

The parameters most relevant to the link budget for the return link appear in table 8.5 below.

Hub antenna diameter	2 m
Hub antenna efficiency	68%
VSAT antenna diameter	0.8 m
VSAT antenna efficiency	60%
VSAT power	0.8 W
Satellite G/T at <i>Ku</i> band	8 dBi/K
Satellite G/T at <i>V</i> band T_s of 900K	16.7 dBi/K
Satellite system noise temperature at <i>Ku</i> band	500 K
Earth station system noise temperature at <i>Ku</i> band	150 K
Satellite system noise temperature at <i>V</i> band	900 K
Earth station system noise temperature at <i>V</i> band	350 K
Satellite EIRP at <i>Ku</i> band	51 dBW
Satellite EIRP at <i>V</i> band	61.8 dBW
Number of carriers at <i>Ku</i> band	3
Number of carriers at <i>V</i> band	12
Burst rate of each <i>Ku</i> band carrier (QPSK modulation, 1/2 rate coding, 5 μ s guard time)	679.1 kbps
Burst rate of each <i>V</i> band carrier (QPSK modulation, 6/7 rate coding, 5 μ s guard time)	397.2 kbps
Occupied <i>Ku</i> band transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	1.495 MHz
Occupied <i>V</i> band transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	3.697 MHz
Minimum C/N for BER of 10^{-6} at <i>Ku</i> band	5.51 dB
Minimum C/N for BER of 10^{-6} at <i>V</i> band	8.51 dB
Total information bit rate available for 58 stations at 64 kbps	3.71 Mbps
Total information bit rate including ATM header overhead	4.92 Mbps
<i>Ku</i> band uplink frequency	13.87 GHz
<i>Ku</i> band downlink frequency	11.35 GHz
<i>V</i> band uplink frequency	50.42 GHz
<i>V</i> band downlink frequency	39.52 GHz
Miscellaneous losses	0.5 dB
Gaseous losses at <i>Ku</i> band uplink frequency	0.7 dB
Gaseous losses at <i>Ku</i> band downlink frequency	0.5 dB
Gaseous losses at <i>V</i> band uplink frequency	3 dB
Gaseous losses at <i>V</i> band downlink frequency	1 dB

Table 8.5: Most significant link budget parameters for the frequency diversity return link

Again, two versions of the scheme were investigated, one with site diversity and one without. A flow diagram of the Matlab program is provided in figure 8.12 and the Matlab code can be found in appendix S.

Here the case where more stations require resources than those provided is referred to as *Ku* band exhaustion, rather than frame exhaustion, which implies a variable duration burst.

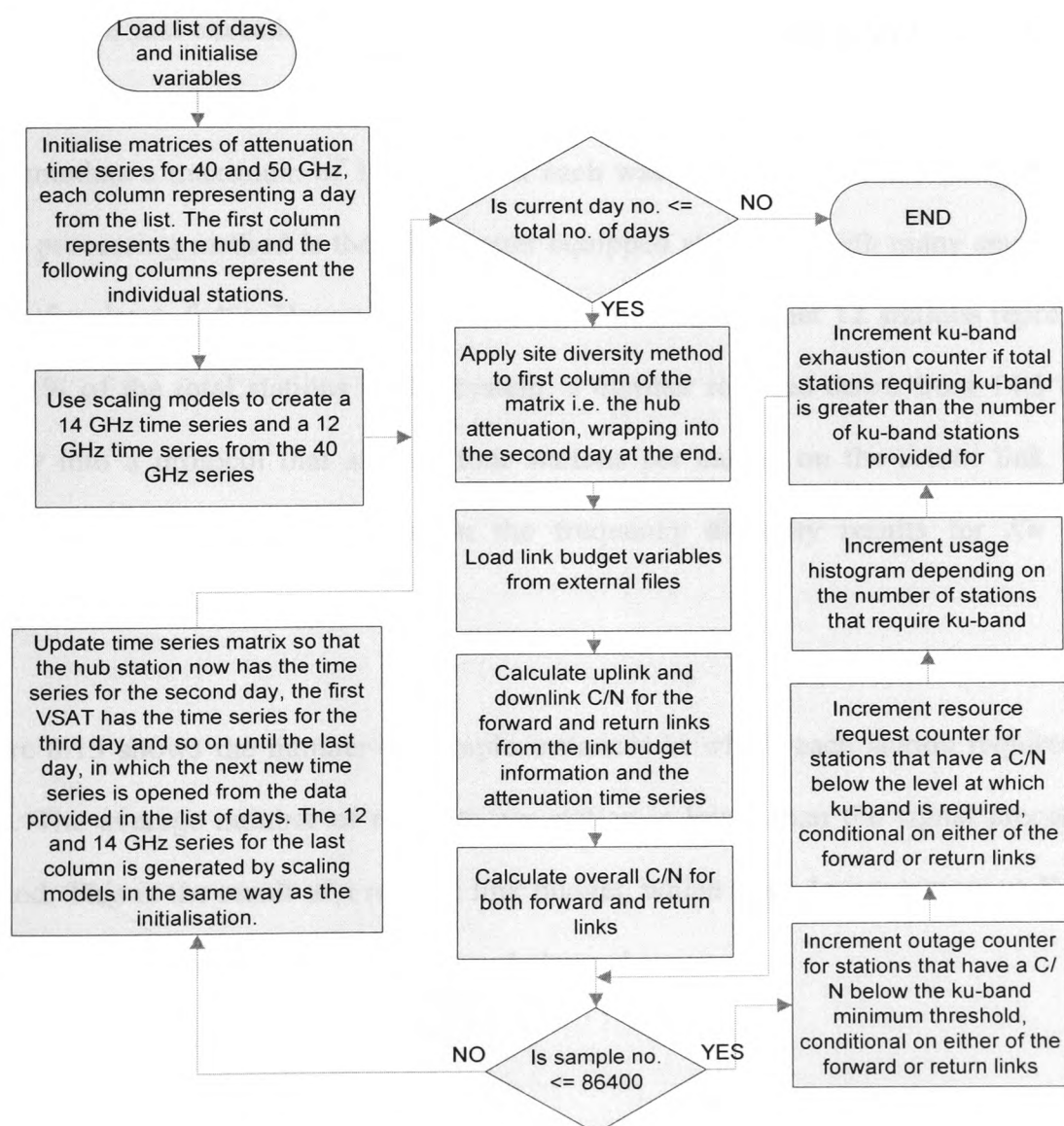


Figure 8.12: Flow diagram of Matlab program of Frequency Diversity FMT with Site Diversity

The frequency diversity scenario has 160497 samples in which more stations require *Ku* band than there are *Ku* band stations available. This equates to a *Ku* band exhaustion probability of 0.963%. This is an increase of 0.138% compared with the signal processing method. This is in part due to the incremental nature of the signal processing method in contrast to the fixed nature of the frequency diversity method. Although the frequency diversity method gives each station a larger maximum fading range, it only accommodates a constant number of stations (12 in this case) regardless of whether the stations have just exceeded the threshold requirement to warrant a *Ku* band slot or are near the outage threshold. The signal processing method on the other hand could accommodate a maximum of 18 stations if each was in the lowest FMT category. The signal processing method is therefore better equipped at dealing with many small fades than a few large fades. However, it should also be stressed that 12 stations represents just 21% of the total stations in the system, a number rounded down from 14.5 to fit exactly into a protocol that assigns four stations per carrier on the return link. This obviously has a negative impact on the frequency diversity results for *Ku* band exhaustion probability.

Figure 8.13 shows the number of sample instances in which each station required *Ku* band. The average number of requests per station is lower than the signal processing method. This is the result of a revised link budget, which has a larger margin at *V* band due to some traffic occupying the *Ku* band channel.

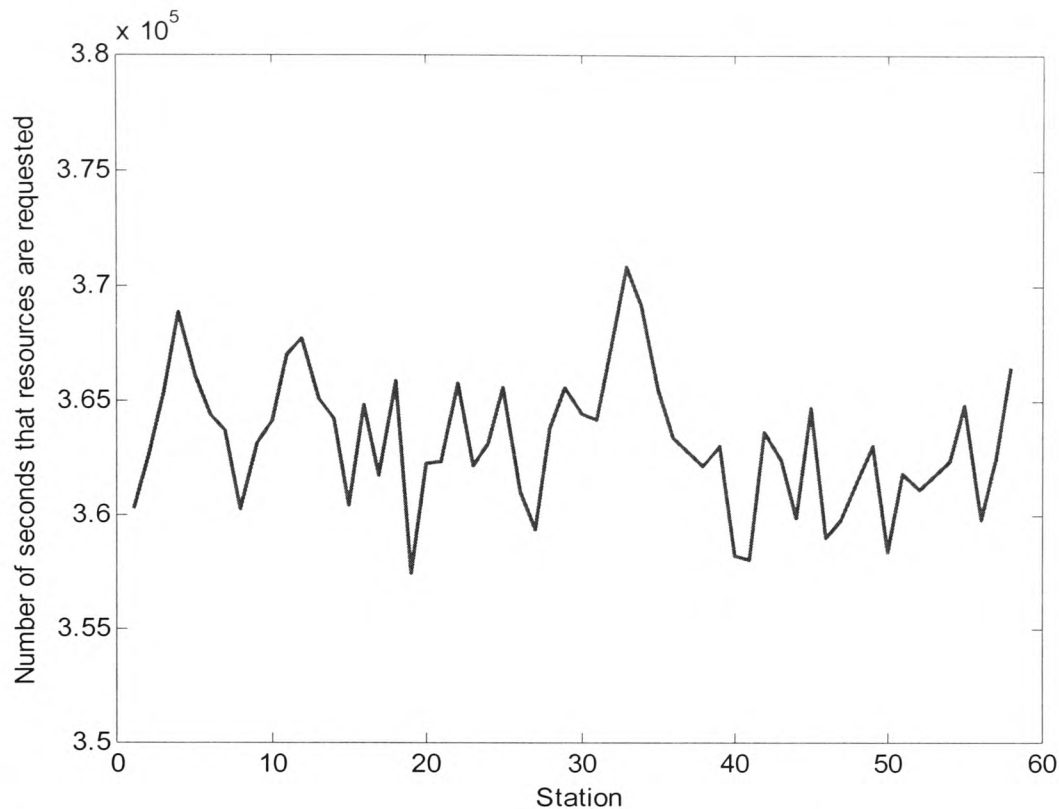


Figure 8.13: Number of resource requests per station for frequency diversity scheme with site diversity

Figure 8.14 shows the number of sample instances in which each station is in outage, i.e. the attenuation at Ku band is too high for the system to operate at the desired BER. The mean number of outages per station is much lower than the signal processing method and can be explained by the larger fading range provided by Ku band when compared with that of the signal processing scenario. Although this graph would benefit from a smaller vertical axis, it has been shown in this way to be immediately comparable with figure 8.3.

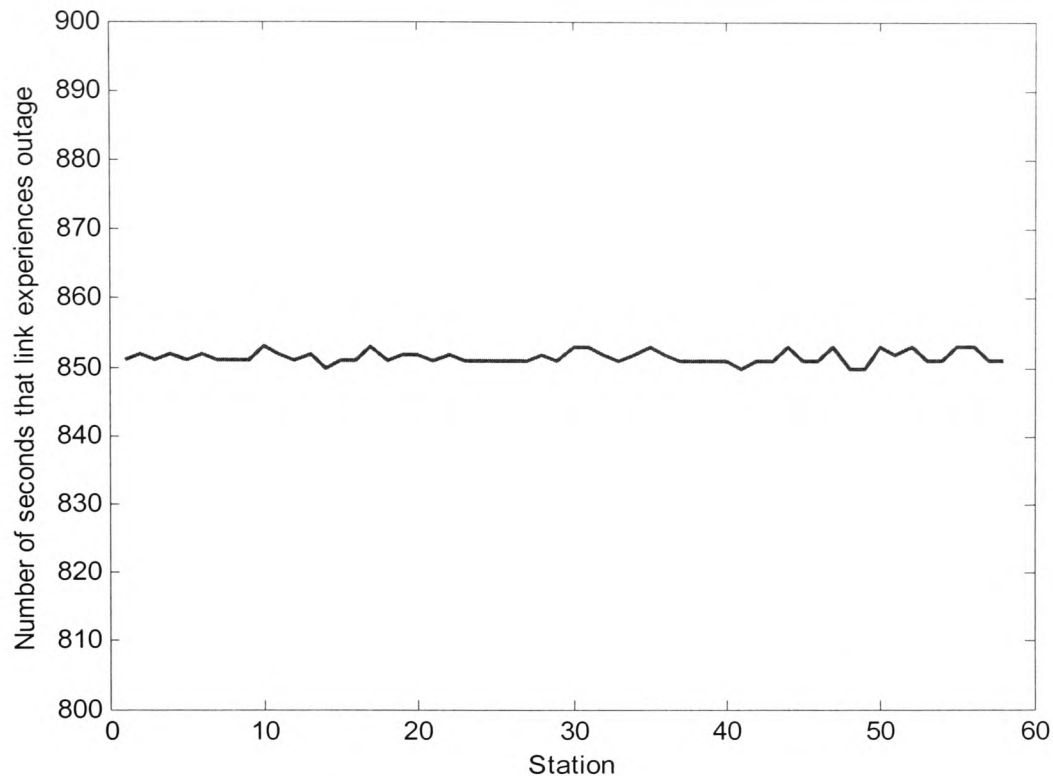


Figure 8.14: Number of seconds in outage per station for frequency diversity scheme with site diversity

Figure 8.15 shows the probability density of the total number of *Ku* band stations that would be required for all stations to be able to communicate at the designated BER. The density is highest at zero but is also high until about 5 stations, where it levels. At 57 and 58 stations (approaching 500% of resources provided) there is a sudden increase in the density which is related to the hub fading. Sometimes, despite there being site diversity, the hub will experience fading. In this case, every station could need FMT protection, depending on the severity of the hub fade and the VSATs own fading level. Obviously, if 58 stations were *Ku* band then the system would simply be a *Ku* band system with a fixed margin, but this graph of probability density function helps to understand the cumulative distribution function in the following graph. Unlike the signal processing scenario, here the combined result for both the forward and return

links is shown. Since the number of *Ku* band channels is fixed there is little point in showing both forward and return links as there is no dynamic component to compare as with the frame duration in the signal processing scheme, i.e. if a station utilizes *Ku* band on the forward link it also utilizes it on the return link.

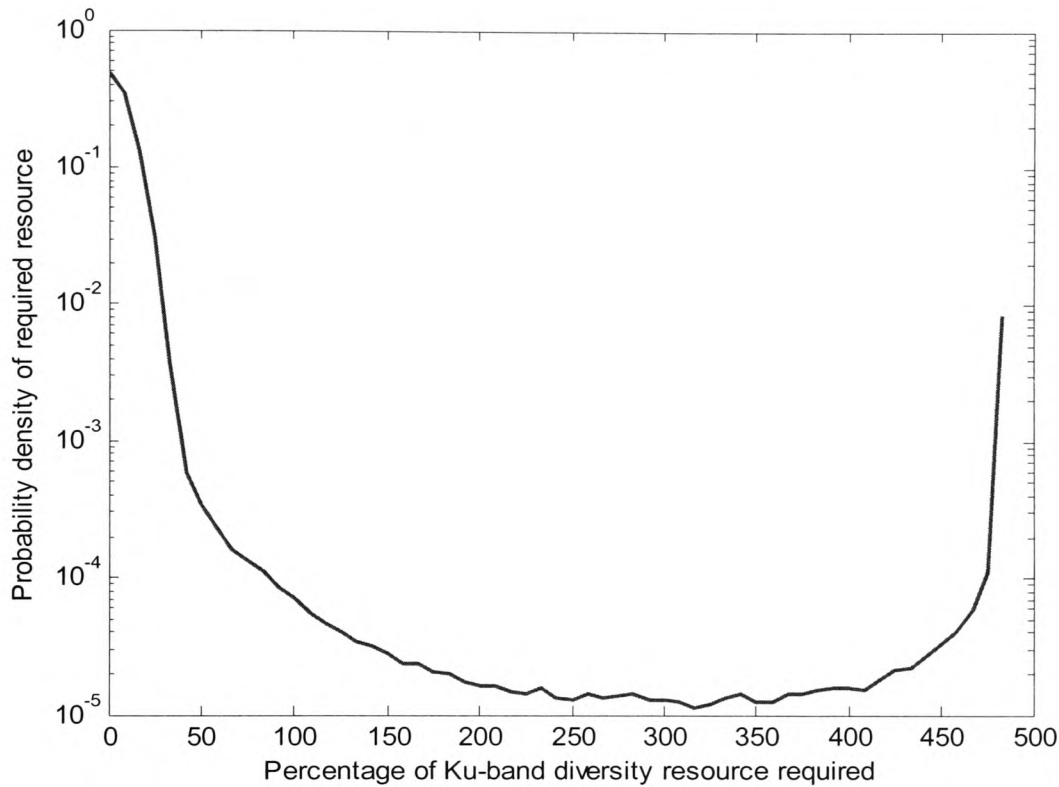


Figure 8.15: PDF of percentage of *Ku* band resource required for all stations to transmit at the chosen BER with site diversity

Figure 8.16 shows the cumulative distribution function up to 200% of the resources provided. Beyond about 42% of the resources provided (5 stations) there is virtually no improvement in the *Ku* band exhaustion probability. This is due to the dominance of the hub fading; the percentage at which hub fades cause resource requests from the VSATs is the minimum outage percentage possible in the system. This figure also shows that 5 stations at *Ku* band represent the optimum use of resources.

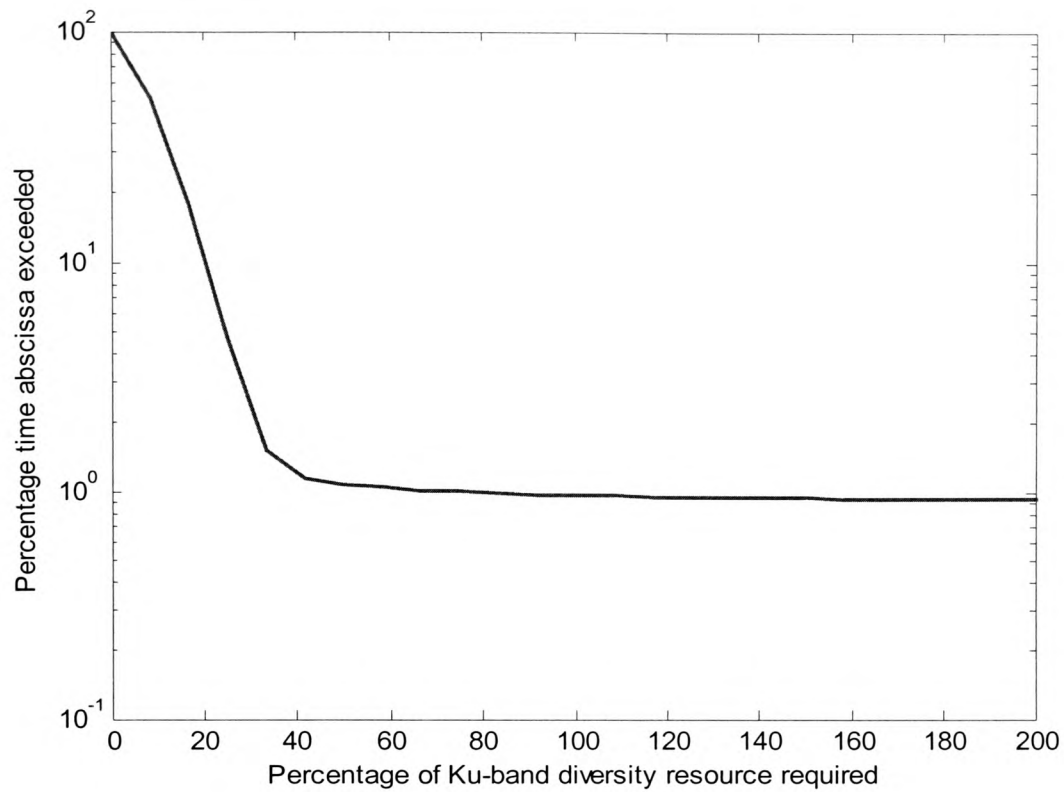


Figure 8.16: CDF of percentage of *Ku* band resource required for all stations to transmit at the desired BER with site diversity

Another scenario was investigated where no site diversity was included but the results added no further insight than those from the signal processing scenario. All aspects of the system were degraded sufficiently to make it an unfeasible solution. For brevity the results have not been included.

8.5 *Hybrid FMT Scenario*

The results for the signal processing and frequency diversity schemes reveal that the two FMTs may complement each other well since the signal processing case is able to assign resources more efficiently with several fading thresholds whereas the frequency diversity scheme has a larger overall dynamic range. Consequently, a hybrid FMT scenario was designed with four stations accommodated at *Ku* band. This required a separate *Ku* band channel on the forward link designed to accommodate four stations with fixed durations and one of fifteen carriers on the return link to be at *Ku* band with fixed durations. Four stations (or 6.7% of the capacity) were chosen as this fits exactly into the protocol for the return channel, four stations being the number per carrier. The remaining 18.3% spare capacity was provided using the signal processing method. Although previous results have already shown that this is larger than the optimum of 10%, the same overall spare capacity was used to enable a useful comparison between methods.

The parameters most relevant to the link budget for the hybrid scenario forward link appear in table 8.6 below. The same assumptions of hub and VSAT locations are equally valid for this scenario as for the previous two. The signal processing FMT uses the same format as that which is shown in table 8.1, albeit with a smaller reserve duration.

Hub antenna diameter	2 m
Hub antenna efficiency	68%
VSAT antenna diameter	0.8 m
VSAT antenna efficiency	60%
Hub power	5 W
Satellite G/T at <i>Ku</i> band	8 dBi/K
Satellite G/T at <i>V</i> band T_s of 900K	16.7 dBi/K
Satellite system noise temperature at <i>Ku</i> band	500 K
Earth station system noise temperature at <i>Ku</i> band	150 K
Satellite system noise temperature at <i>V</i> band	900 K
Earth station system noise temperature at <i>V</i> band	350 K
Satellite EIRP at <i>Ku</i> band	51 dBW
Satellite EIRP at <i>V</i> band	61.8 dBW
Number of carriers at <i>Ku</i> band	1
Number of carriers at <i>V</i> band	1
Burst rate of each <i>Ku</i> band carrier (QPSK modulation, 1/2 rate coding, 5 μ s guard time)	4.584 Mbps
Burst rate of each <i>V</i> band carrier (QPSK modulation, 6/7 rate coding, 5 μ s guard time)	44.89 Mbps
Occupied <i>Ku</i> band transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	3.094 MHz
Occupied <i>V</i> band transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	30.30 MHz
Minimum C/N for BER of 10^{-6} at <i>Ku</i> band	5.51 dB
Minimum C/N for BER of 10^{-6} at <i>V</i> band	8.51 dB
Total information bit rate available for 58 stations at 512 kbps	29.7 Mbps
Total information bit rate including ATM header overhead	33.2 Mbps
<i>Ku</i> band uplink frequency	13.87 GHz
<i>Ku</i> band downlink frequency	11.35 GHz
<i>V</i> band uplink frequency	50.42 GHz
<i>V</i> band downlink frequency	39.52 GHz
Miscellaneous losses	0.5 dB
Gaseous losses at <i>Ku</i> band uplink frequency	0.7 dB
Gaseous losses at <i>Ku</i> band downlink frequency	0.5 dB
Gaseous losses at <i>V</i> band uplink frequency	3 dB
Gaseous losses at <i>V</i> band downlink frequency	1 dB

Table 8.6: Most significant link budget parameters for the hybrid FMT forward link

The parameters most relevant to the return link for the hybrid FMT scenario appear in table 6.7 below. Again, the assumptions of hub and VSAT location apply, as does the format for the signal processing scenario that appears in table 8.1.

Hub antenna diameter	2 m
Hub antenna efficiency	68%
VSAT antenna diameter	0.8 m
VSAT antenna efficiency	60%
VSAT power	0.8 W
Satellite G/T at <i>Ku</i> band	8 dBi/K
Satellite G/T at <i>V</i> band T_s of 900K	16.7 dBi/K
Satellite system noise temperature at <i>Ku</i> band	500 K
Earth station system noise temperature at <i>Ku</i> band	150 K
Satellite system noise temperature at <i>V</i> band	900 K
Earth station system noise temperature at <i>V</i> band	350 K
Satellite EIRP at <i>Ku</i> band	51 dBW
Satellite EIRP at <i>V</i> band	61.8 dBW
Number of carriers at <i>Ku</i> band	1
Number of carriers at <i>V</i> band	17
Burst rate of each <i>Ku</i> band carrier (QPSK modulation, 1/2 rate coding, 5 μ s guard time)	679.1 kbps
Burst rate of each <i>V</i> band carrier (QPSK modulation, 6/7 rate coding, 5 μ s guard time)	397.2 kbps
Occupied <i>Ku</i> band transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	1.495 MHz
Occupied <i>V</i> band transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	4.314 MHz
Minimum C/N for BER of 10^{-6} at <i>Ku</i> band	5.51 dB
Minimum C/N for BER of 10^{-6} at <i>V</i> band	8.51 dB
Total information bit rate available for 58 stations at 64 kbps	3.71 Mbps
Total information bit rate including ATM header overhead	4.92 Mbps
<i>Ku</i> band uplink frequency	13.87 GHz
<i>Ku</i> band downlink frequency	11.35 GHz
<i>V</i> band uplink frequency	50.42 GHz
<i>V</i> band downlink frequency	39.52 GHz
Miscellaneous losses	0.5 dB
Gaseous losses at <i>Ku</i> band uplink frequency	0.7 dB
Gaseous losses at <i>Ku</i> band downlink frequency	0.5 dB
Gaseous losses at <i>V</i> band uplink frequency	3 dB
Gaseous losses at <i>V</i> band downlink frequency	1 dB

Table 8.7: Most significant link budget parameters for the hybrid FMT return link

Figure 8.17 shows the flow diagram of the Matlab program used to gather statistics for the hybrid scenario and the Matlab code can be found in appendix T.

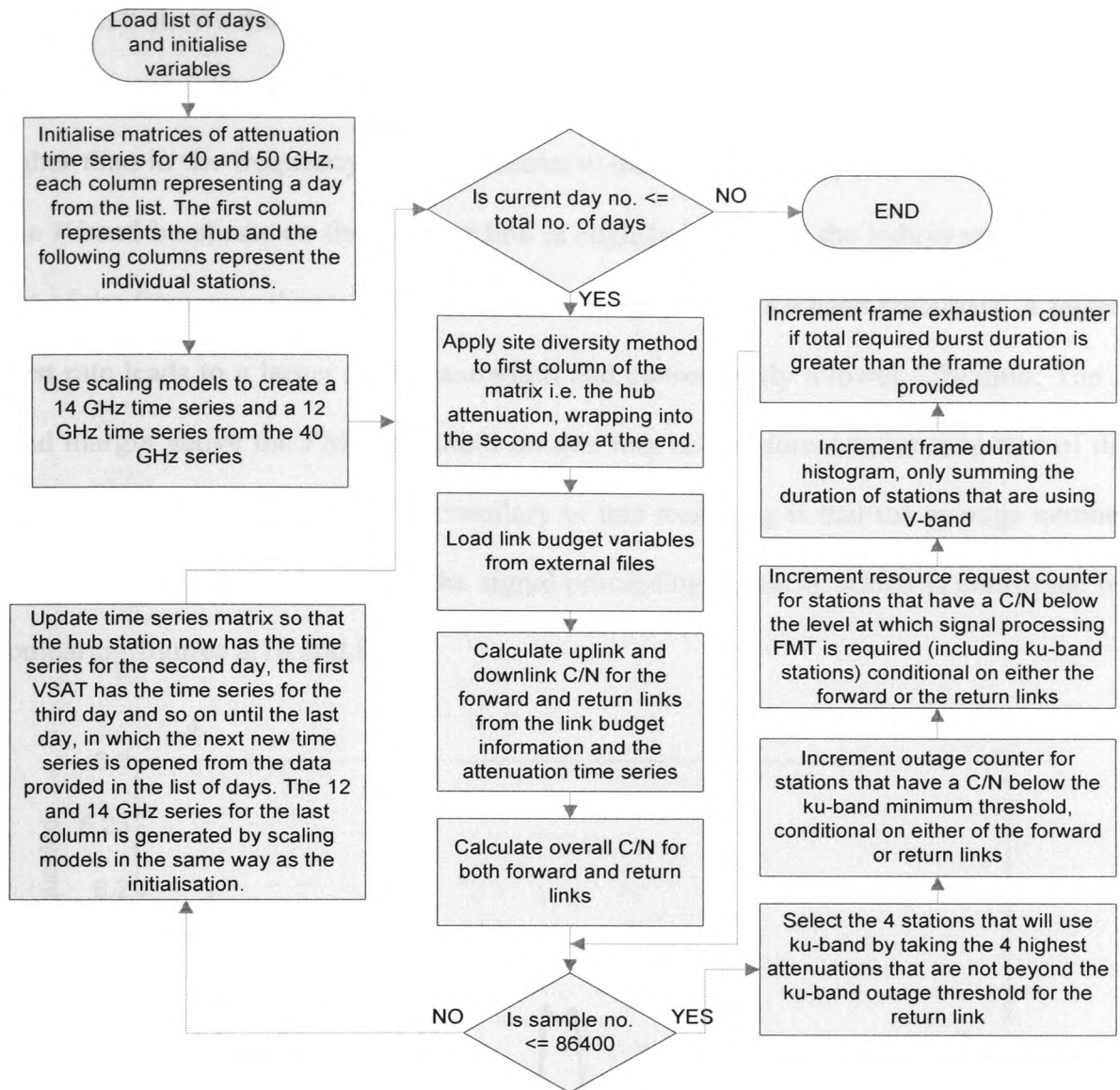


Figure 8.17: Flow diagram of Matlab program of hybrid FMT with site diversity

This scenario has 143146 samples that would result in frame exhaustion. This equals a frame exhaustion probability of 0.858%, fractionally more than the signal processing method and resulting from the fact that the signal processing technique is now limited to a shorter reserve frame duration, since some of the spare resources have been allocated in *Ku* band.

Figure 8.18 shows the number of resource requests per station, including those that have been assigned a *Ku* band carrier. The average number of requests per station is slightly higher than in the frequency diversity scenario due to the slightly different link budget. The *V* band burst rate on the forward link is slightly higher for the hybrid scenario than that of the frequency diversity scenario since there are less *Ku* band resources. A higher burst rate leads to a larger noise bandwidth and consequently a lower C/N ratio. The *V* band margin above the FMT threshold on this link is therefore smaller than that of the frequency diversity scenario. The corollary of this reasoning is that the average number of requests should be lower than the signal processing scenario, which is confirmed by comparing figures 8.18 and 8.2.

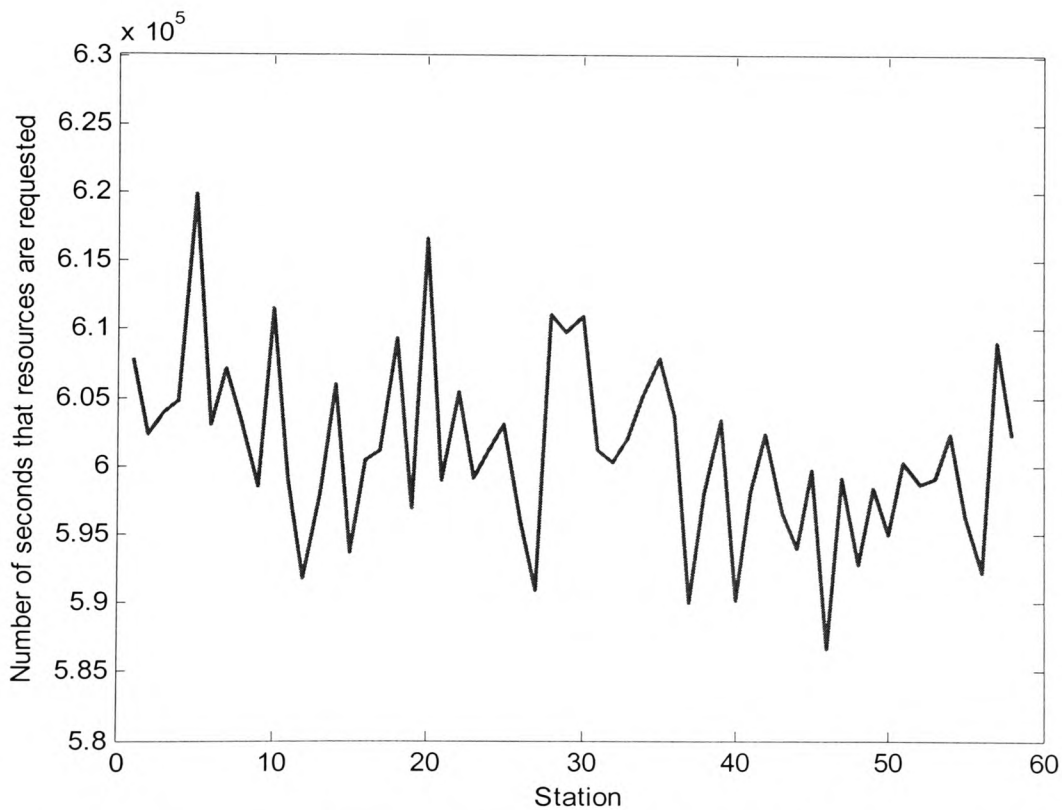


Figure 8.18: Number of resource requests per station for hybrid FMT scheme with site diversity

Figure 8.19 shows the number of outages per station, including those that are assigned Ku band. This is simply a count of the instances that each station has a C/N low enough on the Ku band channel to prohibit communication at the specified BER. The average number of outages per station is slightly less than that of the frequency diversity scenario (842 compared with 852) due to the lower burst rate on the Ku band channel.

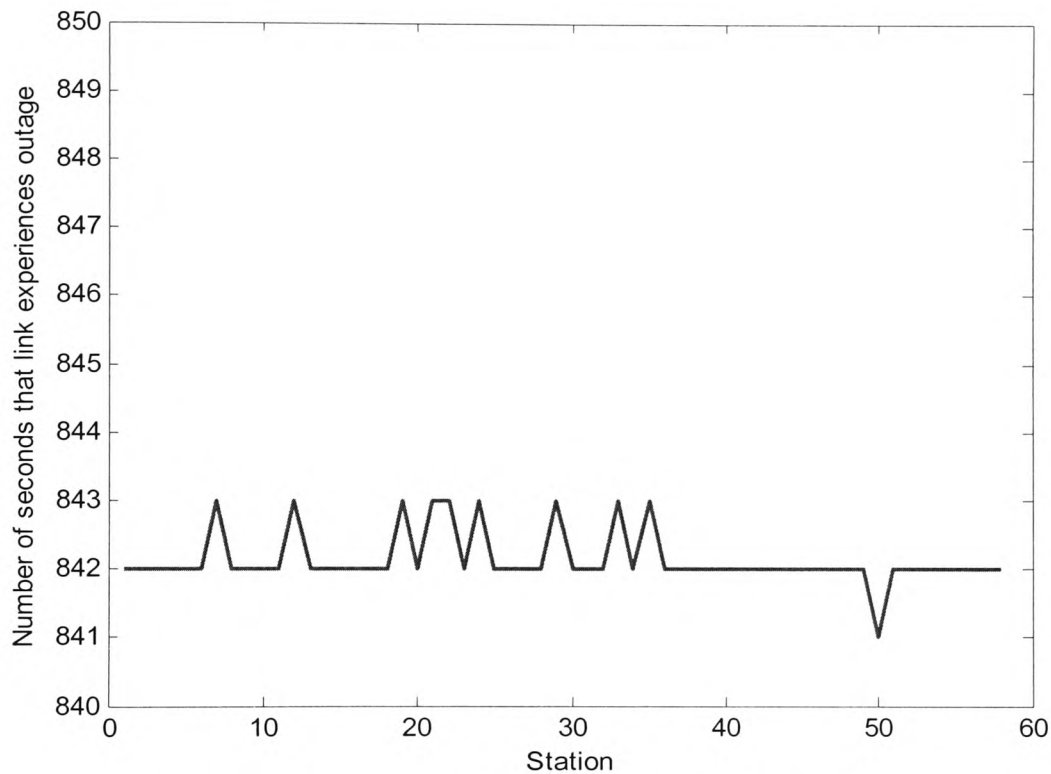


Figure 8.19: Number of seconds in outage per station for hybrid FMT scheme with site diversity

Figure 8.20 shows the probability density of the frame duration required for all stations not in outage to transmit with the necessary FMT specifications for the forward link.

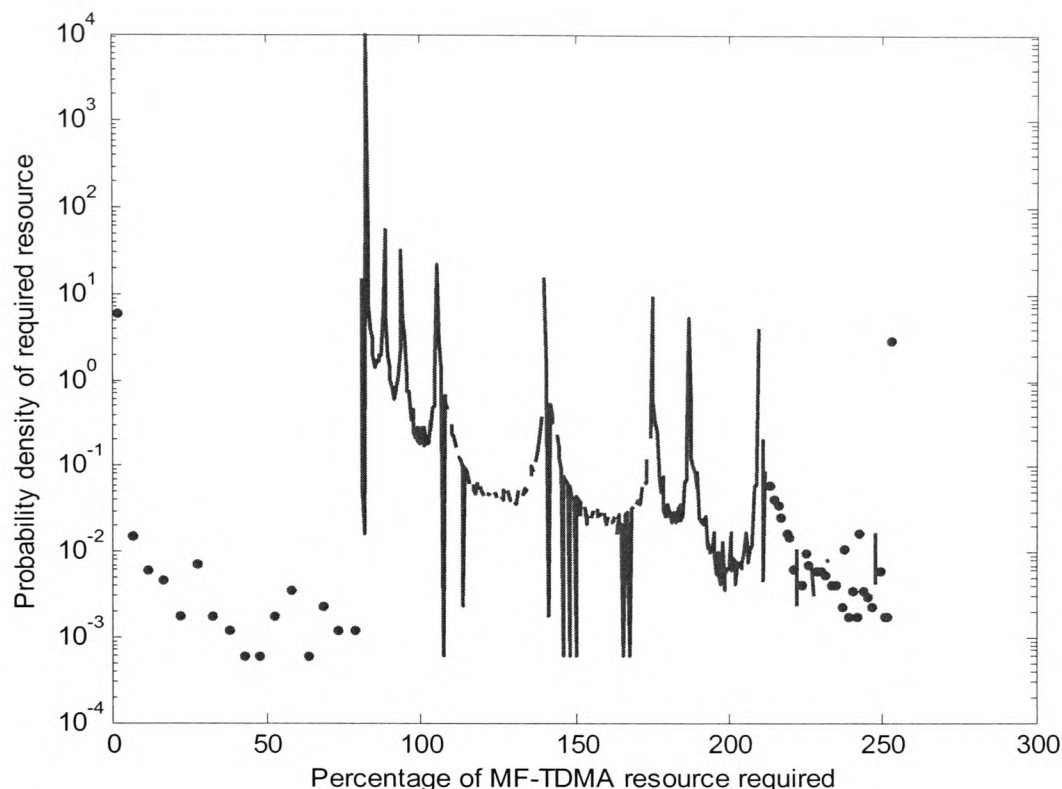


Figure 8.20: PDF of percentage of MF-TDMA frame duration required for all stations to transmit at the chosen BER on the forward link of the hybrid FMT scenario with site diversity

Figure 8.21 shows the cumulative distribution of the frame duration required for all stations not in outage to transmit with the necessary FMT specifications for the forward link. The frame exhaustion probability drops even more rapidly from the nominal value than for the signal processing method alone, due to the fact that the four stations with the most severe fading that are not already in outage are accommodated in the *Ku* band. For a system that is roughly equivalent to the signal processing scenario in terms of frame exhaustion probability (a probability of about 2%), a reserve frame duration of merely 1.8% or 0.36 ms is sufficient. If considered in combination, this amounts to about 8.5% spare capacity altogether, slightly lower than that of the signal processing scenario. However this kind of combination is not very meaningful since the link

budgets are different, the *Ku* band resource requiring more bandwidth per station due to the higher code rate employed.

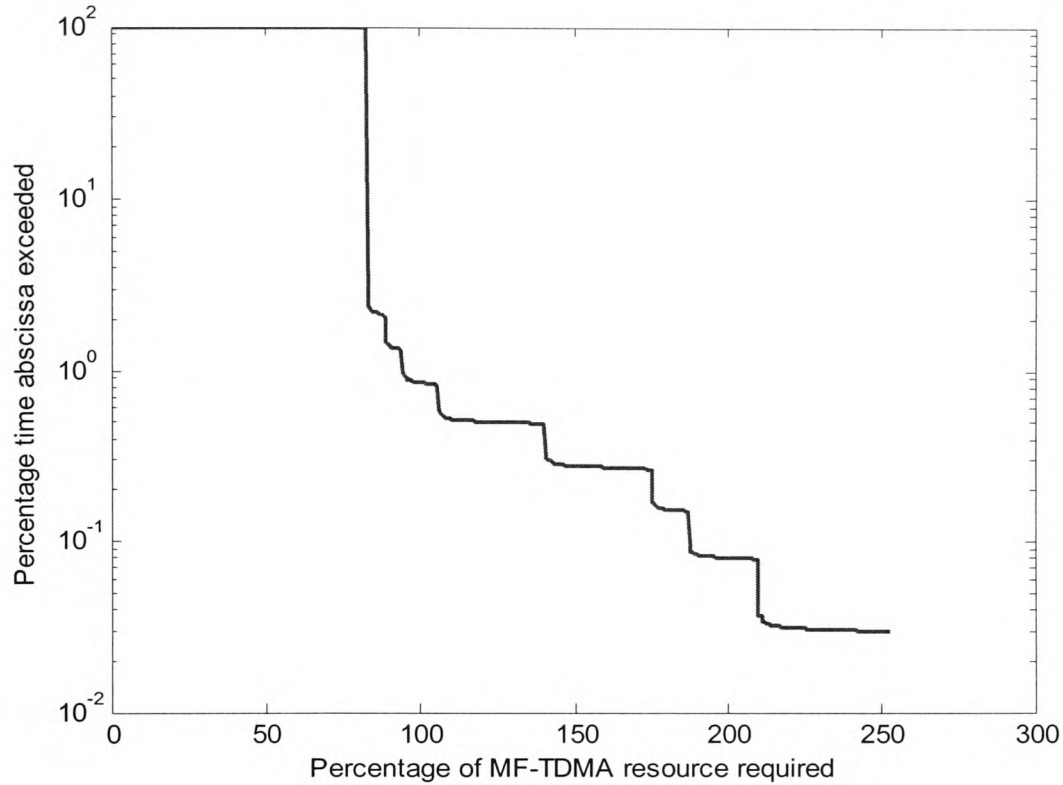


Figure 8.21: CDF of percentage of MF-TDMA frame required for all stations to transmit at the desired BER on the forward link of the hybrid FMT scenario with site diversity

Figure 8.22 shows the probability density of the frame duration required for all stations not in outage to transmit with the necessary FMT specifications for the return link.

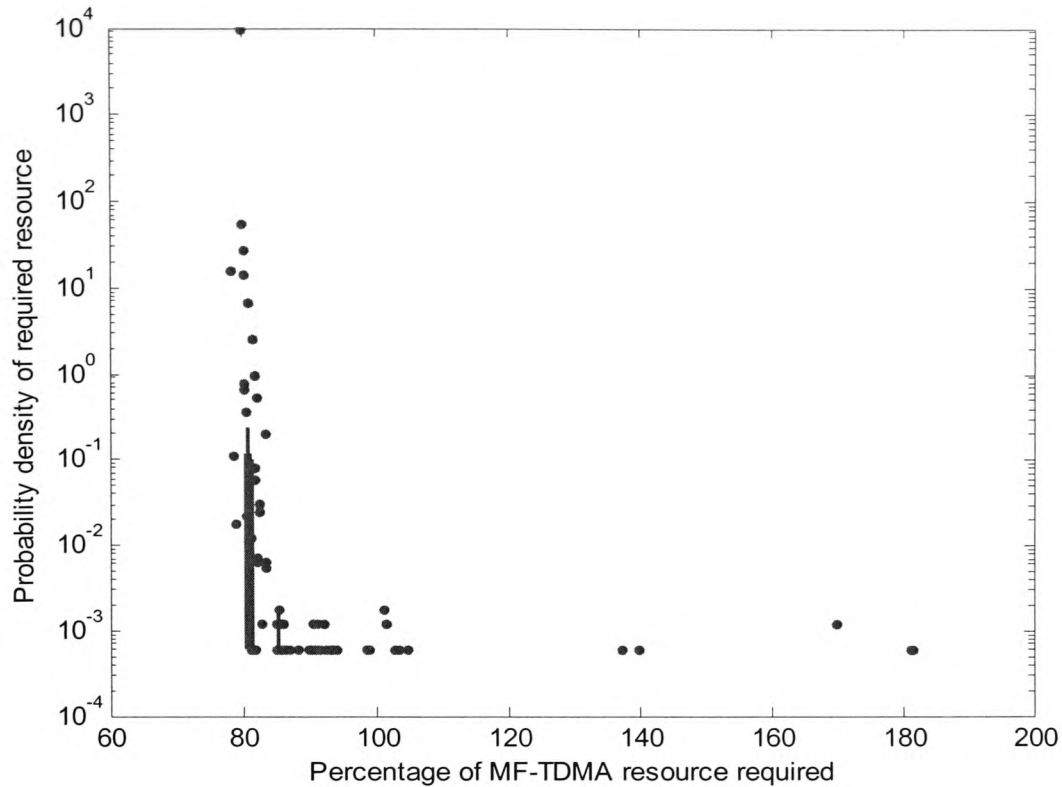


Figure 8.22: PDF of percentage of MF-TDMA frame duration required for all stations to transmit at the chosen BER on the return link of the hybrid FMT scenario with site diversity

Figure 8.23 shows the cumulative distribution of the frame duration required for all stations not in outage to transmit with the necessary FMT specifications for the return link. In this case to achieve a frame exhaustion probability roughly equal to the signal processing method (about 0.01%) a spare resource of 2.8% is required. This equates to just 39% of a single carrier, indicating that perhaps either the *Ku* band channel or the spare carriers on the *V* band channel should be redimensioned from carrying 4 stations to carrying 1 or 2 stations.

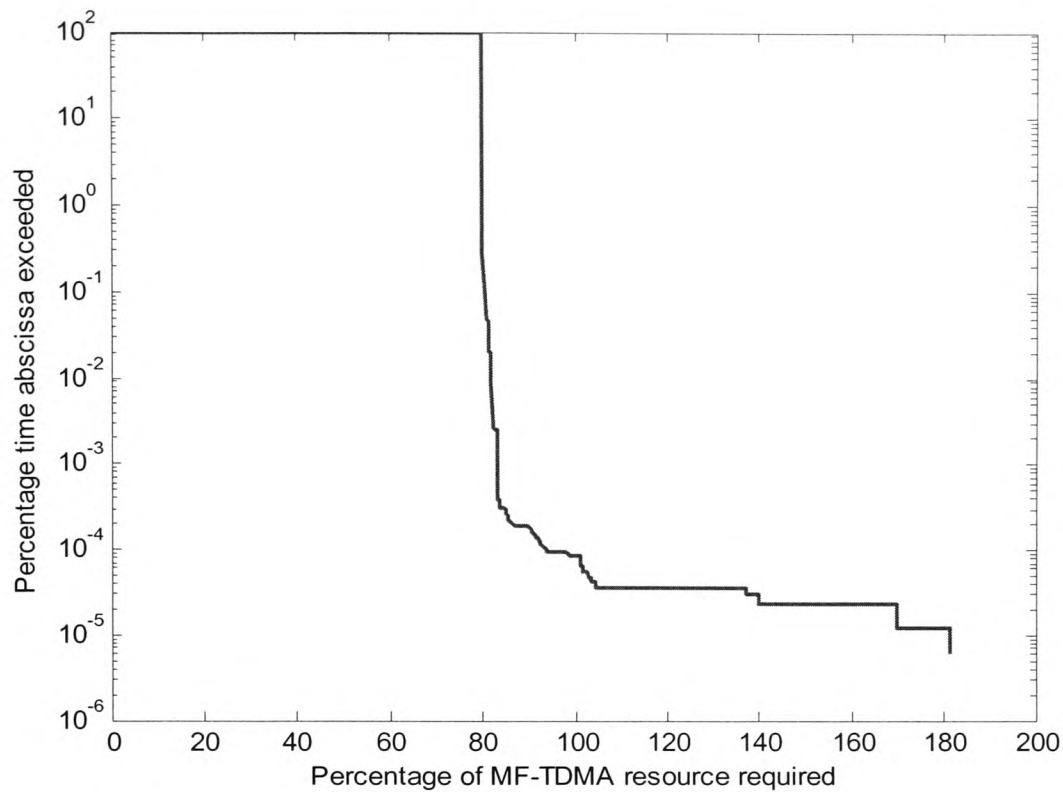


Figure 8.23: CDF of percentage of MF-TDMA frame required for all stations to transmit at the desired BER on the return link of the hybrid FMT scenario with site diversity

A scenario without site diversity was also investigated but has been omitted for brevity since it provided no significant new information.

8.6 Conclusions

The investigation of the three different scenarios has led to several significant conclusions which contributed to the final design stage.

Most importantly, it is clear that the frequency diversity and signal processing schemes are complementary. The frequency diversity scheme is good at providing a large attenuation margin to a few stations whereas the signal processing scheme is better at providing a small attenuation margin to many stations.

Furthermore, the optimum scheme in terms of performance will be one that utilises both techniques, with perhaps about 10% of extra total resources being allocated to protect against rain-induced fading.

However, the decision to combine FMTs should also include a discussion of the trade-offs involved. Of the three scenarios investigated here, the signal processing scheme would be by far the cheapest as it does not require two sets of RF equipment at the receiver. The frequency diversity and hybrid methods both require this and would therefore be a more expensive alternative. However, not only do they perform better for the traffic channel as shown by the results, but they would also immensely benefit the control channel, which would consequently be less affected by outage, as shown in chapter 7. Since the entire system relies on the control channel for the information regarding burst parameters, the probability of error on the control channel should be minimised at all costs. Any error could result in all stations losing communication and a

Chapter 8 – Preliminary Investigation of Combined FMT Performance

recovery mode being initiated. This would of course have a very undesirable effect on the overall outage statistics.

The effect of including site diversity at the hub has also been investigated and it is clear that site diversity applied at the hub on both the forward and return links is an enabling technology. Without it, the rate of outage is simply too high for an acceptable quality of service.

Chapter 9

Final Design, Simulation and Results

9.1 Introduction

In the previous chapter several conclusions were drawn regarding the suitability of different types of FMT together with an estimate of the proportion of resources that should be assigned to overcome rain fades. However, several assumptions were made in order to reduce coding and simulation times. This chapter revises the link budgets according to the information available from the previous chapter and discusses the simulation and results of two FMT scenarios in comparison with a fixed margin system that is standard design practice for *Ku* band and below. Two FMT scenarios are simulated to assess the impact of the proportion of stations that can be accommodated at *Ku* band. Also, two types of analysis were carried out, one in which the physical layer was not implemented in full, but statistics on the availability and frame exhaustion probability were gathered from the C/N ratios over the 193 days of valid data and another in which small sections of the time series were analysed, implementing in full the switching and FEC encoding of the system in order to assess how the system behaved on a short-term basis.

9.2 Design Considerations

Many of the assumptions of the previous chapter were made for time-saving convenience. The following improvements have been made for the simulations described in this chapter:

1. Site diversity switching is no longer instantaneous but takes one second, the lowest resolution possible with a time series sampled at 1 Hz, to take effect.
2. Switching is no longer considered perfect, but a result of the output of a short-term predictor which results in some error when compared with the actual C/N ratio. However, the hybrid predictor discussed in chapter 6 represents a very large increase in processing time than the simpler ADALINE technique, and consequently the ADALINE predictor has been chosen in this case to significantly reduce simulation time. One source of inaccuracy remains, namely the error involved in estimating the C/N ratio from the known bit-sequence in the signal to noise ratio sequence (SS) of the control or traffic channel by the technique discussed in [52].
3. Implementation of the duo-binary turbo code results in a restrictively high processing overhead when considering simulation of the entire data set. However, as mentioned above, two different simulations are carried out here, one that omits the FEC coding and decoding, utilizing only the C/N ratios in gathering annual statistics and another that fully simulates the coding and decoding over short periods within a fading event. The former gives a statistical view of annual outage and frame exhaustion probability in comparison with the case of a fixed link margin system at V band while the latter enables a view of the switching and error performance over periods of seconds to several tens of seconds.
4. A transponder back-off of 6 dB has been included on the return downlink to compensate for interference due to intermodulation products resulting from transponder non-linearity.

5. The control channels are now implemented in full in order to evaluate their annual availability. However, the exact effects of errors occurring within the control channel are not investigated due to the processing overhead involved in fully implementing the hub control actions at a higher network layer.
6. Up-link power control is now implemented with a dynamic range of 6 dB in two 3 dB stages. If the uplink C/N measured by the hub drops by 3 dB, a 3 dBW increase in power is added, and likewise if a 6 dB drop occurs, 6 dBW of extra power is added. The uplink C/N can be interpolated from a lookup table for which a combined figure of the uplink attenuation, downlink attenuation and noise increase from the clear-sky reference is used as an index. A graph of the conversion function between the full C/N reduction and the uplink attenuation for the *Ku* band frequency is shown in figure 9.1. The *V* band uplink attenuation can be scaled from this using the method provided in [9]. A time delay of a second is assumed. This is a pessimistic view of the actual delay but the resolution of the data is 1 second, which limits any delay to this minimum. This is in contrast with the statement in [53] that the measurement of C/N ratio by a closed-loop method is unsuitable for uplink power control, since the uplink attenuation can be estimated from the overall C/N reduction that the hub can measure on the forward link.
7. The traffic described in chapter 7 is now implemented via two look-up tables of communication start and end times over the full 193 day duration for each of the 620 stations. Every time a station begins to transmit, it is allocated the first empty channel of the 58 possible channels. When a station finishes transmitting, it releases the channel it was assigned for use by other stations. Although this utilises the lower channels more than the higher ones, the likelihood of rain

attenuation on the link should still be the same on average, which is a more important consideration than exactly which channel is utilized.

All assumptions made in chapter 8 have been improved considerably, with the exception of perfect division of resources between the MF-TDMA carriers of the return link. This assumption will have the effect of reducing the frame exhaustion probability from the practical case, where occasionally, even though the total frame duration exceeds the duration required, the sharing of the MF-TDMA bursts between all stations is not possible. The omission of an algorithm to assign stations to carriers, maximising the number of stations that can be accommodated, is due to the processing overhead of implementing the hub control algorithm in full as stated previously.

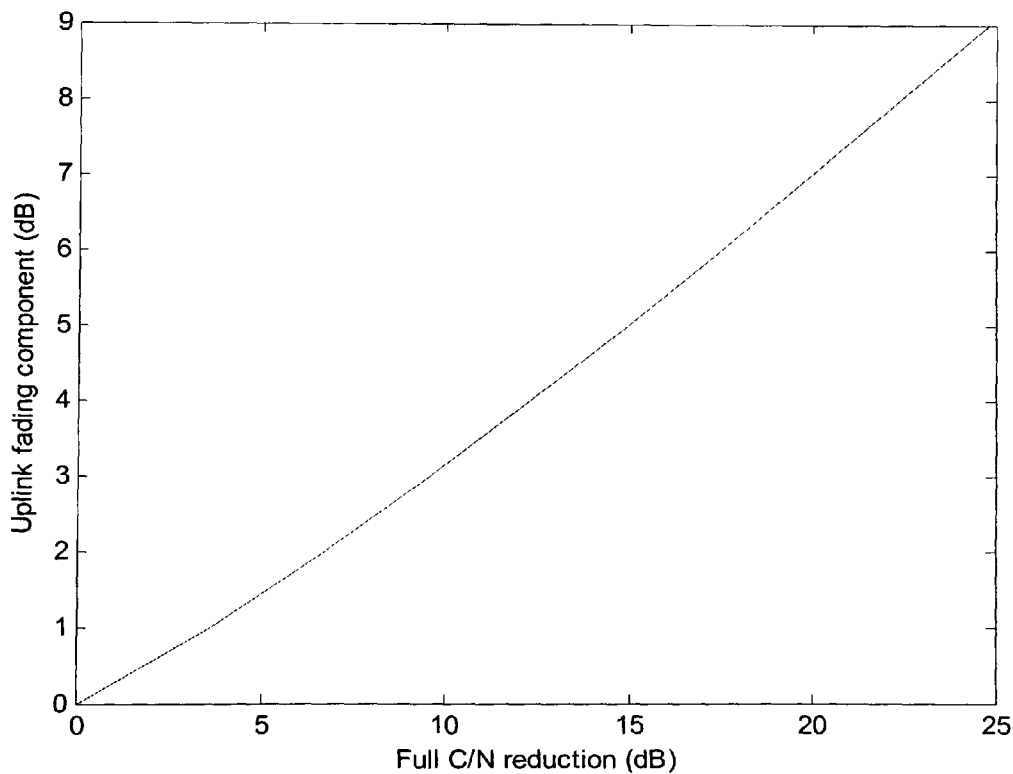


Figure 9.1: Conversion function for Ku band uplink fading from C/N reduction

9.3 Final Design and Results

9.3.1 Software Approach

A flow diagram of the statistical analysis is shown in figure 9.2 below, followed by a flow diagram of the short-term switching analysis in figure 9.3. These diagrams are applicable to the two scenarios investigated, the only change being in the link budget parameters loaded at the beginning.

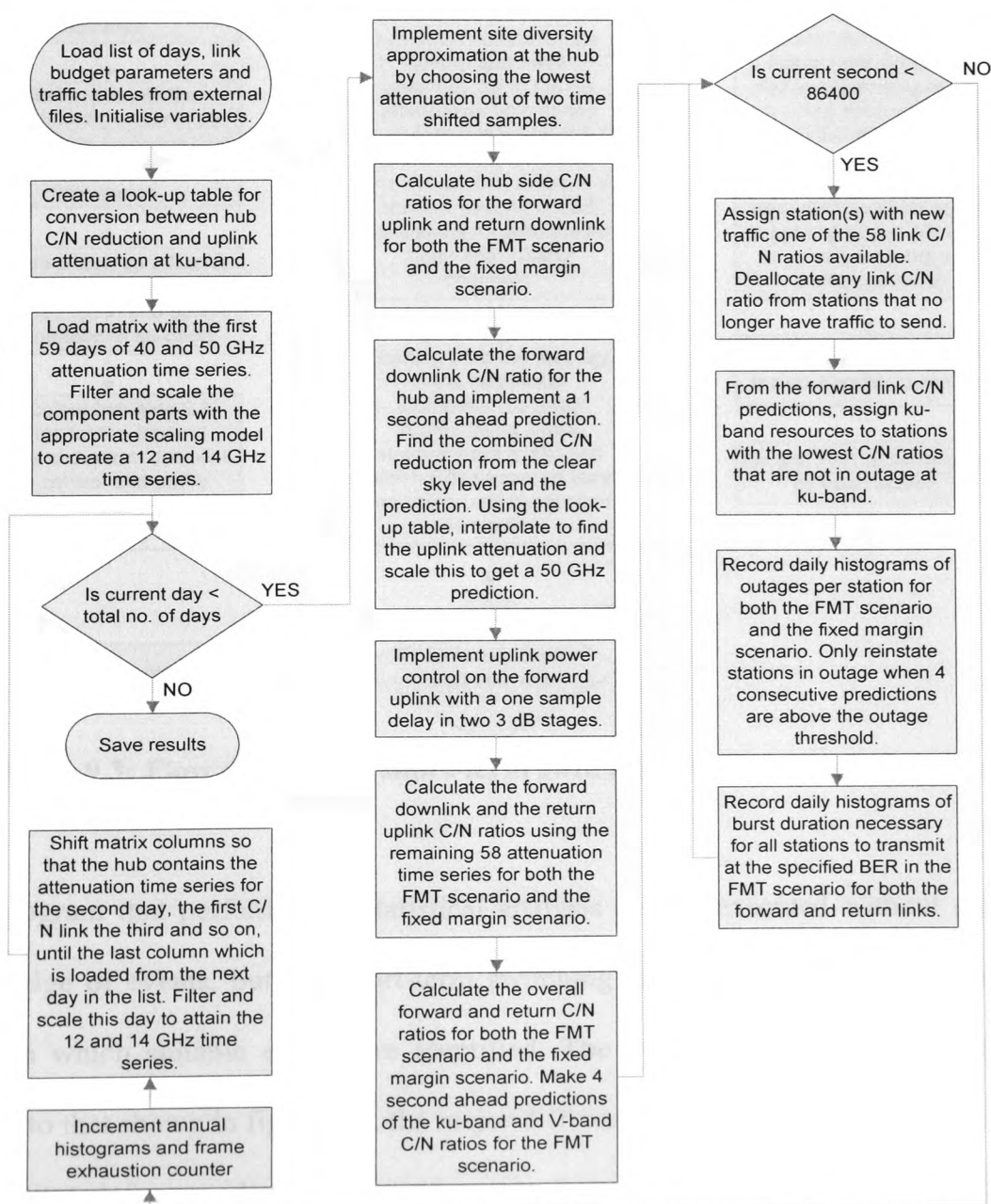
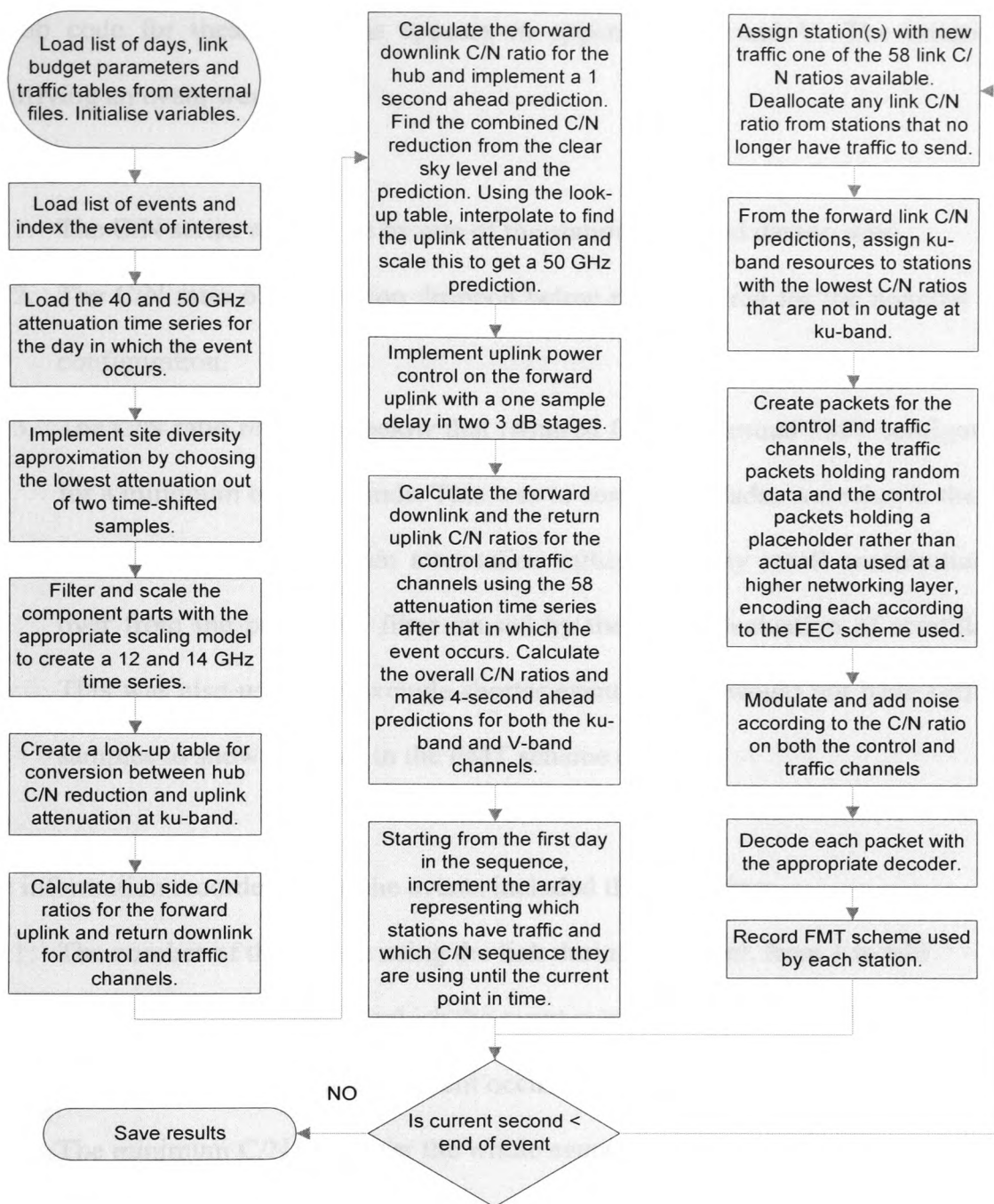


Figure 9.2: Flow diagram of statistical analysis of FMT and fixed margin scenario

Chapter 9 – Final Design, Simulation and Results



The program that performs the statistical analysis can be executed without any prior knowledge of events, but the short-term switching analysis requires a pre-processing stage in which suitable events are identified. The program in which this is done is similar to that shown in figure 9.2, the major difference being that the statistical analysis is substituted for an identification of events for each of the 58 possible links. The

Matlab code for these programs appears in appendices U and V. The criteria for identifying an event were:

1. The C/N series was in use by one of the stations that had data to send.
2. The C/N ratio of the station dropped below that required for the nominal FMT configuration.
3. The C/N ratio remained below that required for the nominal FMT configuration for a minimum of 20 seconds. This was to ensure that fades were due to the slow changing dynamics of rain attenuation rather than any small contribution left over from the predictive filter caused by the rapid fluctuation of scintillation. This was also useful to exclude shorter events which would not have sufficient samples to show changes in the FMT scheme clearly.

The information recorded about the events included the following:

1. The number of the station using the link during the event, from 1 to 620.
2. The number of the day in which the event occurred, from 1 to 193.
3. The C/N series on which the event occurred, from 1 to 58.
4. The minimum C/N ratio over the whole event.
5. The time at which the event began.
6. The time at which the event ended.
7. The number of stations experiencing an event at the point at which the event for this station ended, i.e. the number of concurrent events at the end of this particular event.
8. An array of the C/N series numbers for these concurrent events, from 1 to 58.
9. An array of the station numbers for these concurrent events, from 1 to 620.

10. The maximum number of concurrent events across all stations for the duration of this particular event.

This information was stored in a list of events and analysed separately by another program to identify the periods most likely to show a broad range of FMT configurations, which permitted careful selection of periods over which the FMT switching analysis could be carried out.

9.3.2 Design and Results of a Four-station Frequency Diversity Approach

The first system investigated was that of an FMT incorporating four stations at Ku band with the remaining stations at V band in one of several possible signal processing configurations.

The obvious benefit of this format is that the Ku band return link maintains the same structure as the V band return link, the nominal number of stations at V band being equal.

The amount of resources assigned for the operation of FMTs is 10%, which is the estimate of the best proportion obtained in the previous chapter. This is likely to be larger than necessary since the full resources are rarely used as shown by the traffic PDF in figure 7.6 of chapter 7.

A station can be in any of 10 configurations which are outlined in table 9.1 below.

Configuration Number	Freq. Band	Modulation	Code Rate	C/N Required (dB)	Forward/Return Link Burst Durations (ms)
1	<i>V</i>	QPSK	6/7	8.5	0.357 / 5.317
2	<i>V</i>	QPSK	4/5	7.9	0.382 / 5.660
3	<i>V</i>	QPSK	3/4	7.2	0.407 / 6.022
4	<i>V</i>	QPSK	2/3	6.7	0.457 / 6.727
5	<i>V</i>	QPSK	1/2	5.8	0.608 / 8.862
6	<i>V</i>	BPSK	4/5	4.9	0.763 / 11.32
7	<i>V</i>	BPSK	3/4	4.2	0.814 / 12.04
8	<i>V</i>	BPSK	2/3	3.7	0.914 / 13.45
9	<i>V</i>	BPSK	1/2	2.8	1.217 / 17.72
10	<i>Ku</i>	QPSK	1/2	5.8	4.995 / 4.995

Table 9.1: FMT configurations

The parameters most relevant to the link budget for the forward and return links are shown in tables 9.2 and 9.3 respectively. The assumed locations of the hub and VSATs are the same as discussed in the previous chapter, the majority of Europe being suitable for both.

Hub antenna diameter	2 m
Hub antenna efficiency	68%
VSAT antenna diameter	0.8 m
VSAT antenna efficiency	60%
Hub power (<i>Ku</i> band link)	200 W
Hub power (<i>V</i> band link)	5 W
Satellite G/T at <i>Ku</i> band	8 dBi/K
Satellite G/T at <i>V</i> band T_s of 900K	16.7 dBi/K
Satellite system noise temperature at <i>Ku</i> band	500 K
Earth station system noise temperature at <i>Ku</i> band	150 K
Satellite system noise temperature at <i>V</i> band	900 K
Earth station system noise temperature at <i>V</i> band	350 K
Satellite EIRP at <i>Ku</i> band	51 dBW
Satellite EIRP at <i>V</i> band	61.8 dBW
Number of traffic carriers at <i>Ku</i> band	1
Number of traffic carriers at <i>V</i> band	1
Number of control channel carriers at <i>Ku</i> band	1
Number of stations at <i>Ku</i> band	4
Maximum number of stations at <i>V</i> band	54
Burst rate of each <i>Ku</i> band traffic channel (QPSK modulation, 1/2 rate coding, 5 μ s guard time)	4.584 Mbps
Burst rate of each <i>V</i> band traffic channel (QPSK modulation, 6/7 rate coding, 5 μ s guard time)	37.84 Mbps
Burst rate of control channel (BPSK modulation, 1/2 rate coding, 5 μ s guard time)	110.5 kbps
Occupied <i>Ku</i> band transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	3.094 MHz
Occupied <i>V</i> band transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	25.54 MHz
Occupied <i>Ku</i> band transponder bandwidth for control channel (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	149.1 kHz
Minimum C/N for BER of 10^{-6} at <i>Ku</i> band	5.51 dB
Minimum C/N for BER of 10^{-6} at <i>V</i> band	8.51 dB
Minimum C/N for BER of 10^{-6} for control channel (turbo code)	2.5 dB
Total information bit rate available for 58 stations at 512 kbps	29.7 Mbps
Total information bit rate including ATM header overhead	33.2 Mbps
<i>Ku</i> -band uplink frequency	13.87 GHz
<i>Ku</i> -band downlink frequency	11.35 GHz
<i>V</i> band uplink frequency	50.42 GHz
<i>V</i> band downlink frequency	39.52 GHz
Miscellaneous losses	0.5 dB
Gaseous losses at <i>Ku</i> band uplink frequency	0.7 dB
Gaseous losses at <i>Ku</i> band downlink frequency	0.5 dB
Gaseous losses at <i>V</i> band uplink frequency	3 dB
Gaseous losses at <i>V</i> band downlink frequency	1 dB

Table 9.2: Most significant link budget parameters for the forward link with four stations at *Ku* band

Hub antenna diameter	2 m
Hub antenna efficiency	68%
VSAT antenna diameter	0.8 m
VSAT antenna efficiency	60%
VSAT power (<i>Ku</i> band link)	2 W
VSAT power (<i>V</i> band link)	0.8 W
Satellite G/T at <i>Ku</i> band T_s of 500K	8 dBi/K
Satellite G/T at <i>V</i> band T_s of 900K	16.7 dBi/K
Satellite system noise temperature at <i>Ku</i> band	500 K
Earth station system noise temperature at <i>Ku</i> band	150 K
Satellite system noise temperature at <i>V</i> band	900 K
Earth station system noise temperature at <i>V</i> band	350 K
Satellite EIRP at <i>Ku</i> band	51 dBW
Satellite EIRP at <i>V</i> band	61.8 dBW
Number of traffic carriers at <i>Ku</i> band	1
Number of traffic carriers at <i>V</i> band	16
Number of control channel carriers at <i>Ku</i> band	1
Number of stations at <i>Ku</i> band	4
Maximum number of stations at <i>V</i> band	54
Burst rate of each <i>Ku</i> band traffic channel (QPSK modulation, 1/2 rate coding, 5 μ s guard time)	679.1 kbps
Burst rate of each <i>V</i> band traffic channel (QPSK modulation, 6/7 rate coding, 5 μ s guard time)	397.2 kbps
Burst rate of control channel (QPSK modulation, 1/2 rate coding, 5 μ s guard time)	94.84 kbps
Occupied <i>Ku</i> band transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	498.4 kHz
Occupied <i>V</i> band transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	4.930 MHz
Occupied <i>Ku</i> band transponder bandwidth for control channel (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	64.02 kHz
Minimum C/N for BER of 10^{-6} at <i>Ku</i> band	5.51 dB
Minimum C/N for BER of 10^{-6} at <i>V</i> band	8.51 dB
Minimum C/N for BER of 10^{-6} for control channel (convolutional code)	7.5 dB
Total information bit rate available for 58 stations at 64 kbps	3.71 Mbps
Total information bit rate including ATM header overhead	4.92 Mbps
<i>Ku</i> -band uplink frequency	13.87 GHz
<i>Ku</i> -band downlink frequency	11.35 GHz
<i>V</i> band uplink frequency	50.42 GHz
<i>V</i> band downlink frequency	39.52 GHz
Miscellaneous losses	0.5 dB
Gaseous losses at <i>Ku</i> band uplink frequency	0.7 dB
Gaseous losses at <i>Ku</i> band downlink frequency	0.5 dB
Gaseous losses at <i>V</i> band uplink frequency	3 dB
Gaseous losses at <i>V</i> band downlink frequency	1 dB

Table 9.3: Most significant link budget parameters for the return link with four stations at *Ku* band

The number of seconds that each station was in outage was recorded for each station where outage was defined by the point at which stations could no longer meet QoS at *Ku* band, i.e. the C/N on the *Ku* band traffic channel had dropped below the required C/N. This was then compared with the total transmitting time of each individual station, to arrive at the annual availability of each station. Care was taken in this calculation so that the comparison was made with the amount of time that each station had traffic to send, which was slightly different in each case.

Figure 9.4 shows the availability for each of the 620 stations for both the FMT and fixed margin scenarios. A marked difference between the two methods is immediately apparent. There is a significant variability in annual availability between stations and this is due in part to the low percentage of time that each station has traffic to send, being just 5.4% of the total simulated time on average. However an annual estimate of availability can be made by taking the mean availability of all the stations, which has the same effect as running a single station for a longer period of time. It has already been shown in chapter 8 that temporal decorrelation of events can be substituted for spatial decorrelation of events without introducing any significant deviation in the results with respect to the ITU-R model. The process by which the stations access the resource is further randomised since each station now uses any of 58 possible links, rather than having a single link assigned to it. The results shown in figure 9.4 show no obvious trend and it is therefore assumed that a mean annual availability between stations is representative of the average over a longer period.

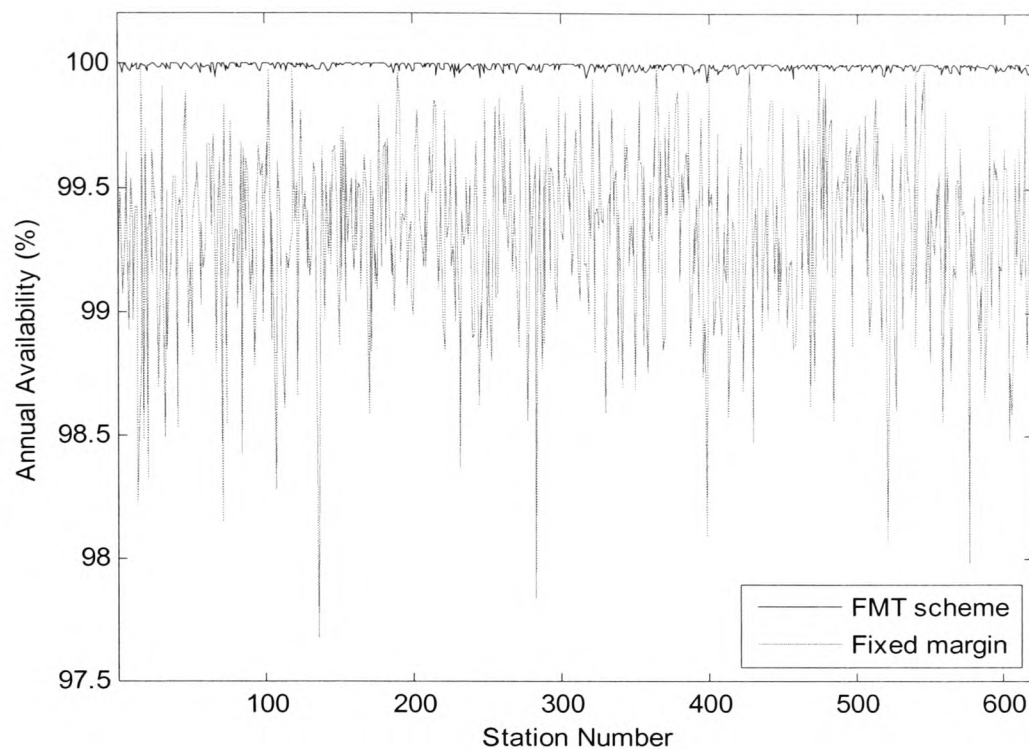


Figure 9.4: Annual availability of stations

The mean annual availability of the FMT scenario is 99.991%, which includes failure due to frame exhaustion in addition to rain outage. The mean annual availability for the fixed margin scenario is 99.273%, revealing a significant difference. In terms of the annual amount of time in outage, the FMT scenario would on average experience outage for almost 46 minutes during a year, while the fixed margin system would experience outage for almost 64 hours during a year. The latter case is unsuitable for the majority of satellite services, not only internet delivery to home or office users, whereas the former is typically acceptable for a wide range of services.

However, there is a further problem with the fixed margin scheme which results from its annual availability having a much larger standard deviation than that of the FMT scheme; the confidence at which QoS parameters can be quoted is reduced dramatically. In the worst case, which can be seen in figure 9.4, several stations have availabilities of

below 98%. This is shown more clearly in figure 9.5, in which the cumulative distribution function of the annual availability of stations is plotted. One can say, for example, with ninety nine percent confidence that service will be available to a station using the FMT scheme for 99.95% of the time whereas for the fixed margin scheme one can only say with ninety nine percent confidence that service will be available for 98.25% of the time. This equates to a drop of 0.04% from the average for the FMT scheme but a 1.023% drop from the average for the fixed margin scheme. This further reinforces the case for an adaptive FMT over a fixed scheme in contrast to the conclusion in [29], that the use of an adaptive FMT scheme is beneficial only for throughput optimization and not availability. Although both these figures may improve over a full year rather than the 193 day subset that has been used, the fixed margin system is likely to remain significantly inferior in this respect as a broader range of rain events are likely to cause outage.

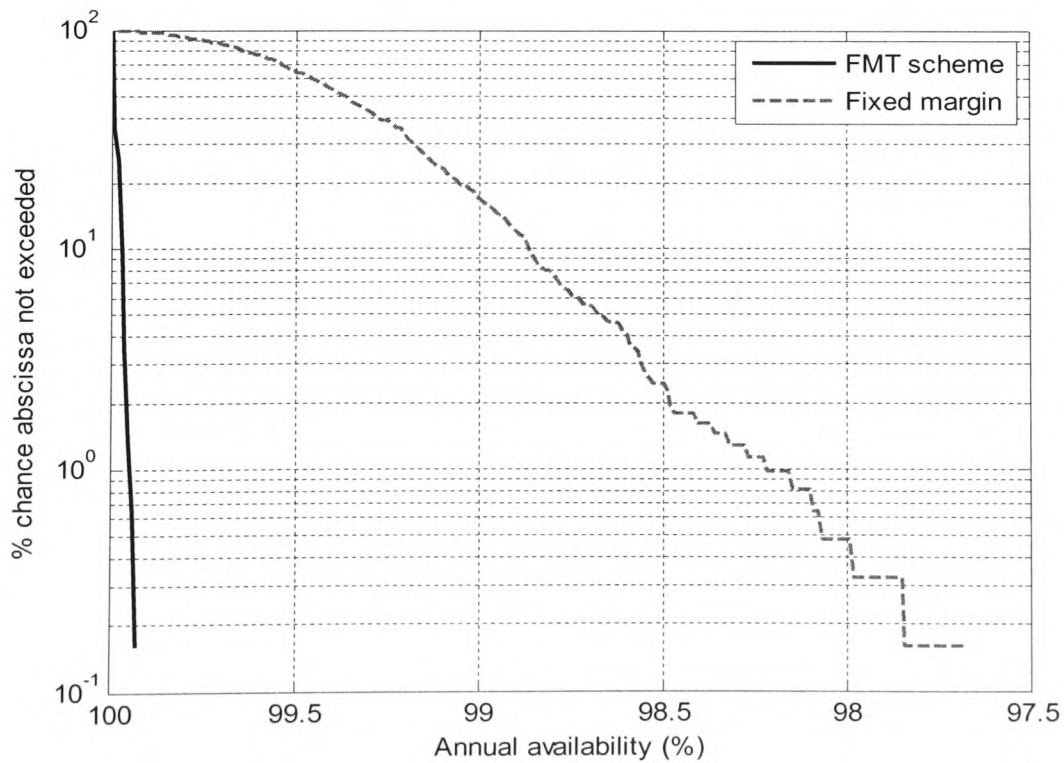


Figure 9.5: CDF of annual availability of stations

Figure 9.6 shows the probability density of the percentage of MF-TDMA resources required for all stations to transmit at the chosen QoS and figure 9.7 shows the corresponding cumulative distribution. These measurements include the total frame duration for all stations with traffic that are not in outage. Frame exhaustion is now very low compared with the results in chapter 8, although this is expected since the MF-TDMA frame is very rarely occupied by the full number of stations that can be accommodated. There are however 202 instances throughout the 193 days in which frame exhaustion occurs, and this is most noticeable in figure 9.7 where the graph extends past the total resources provided but at a very low probability of 4 in 10 million. The MF-TDMA resource that is actually utilized after sufficient stations have been denied resources so that the traffic will fit in the frame is plotted alongside but the difference is so small that it is not visible without magnifying the area of interest. A short example of one such instance appears in figure 9.8, which shows a hub fading event on the 8th August 1998.

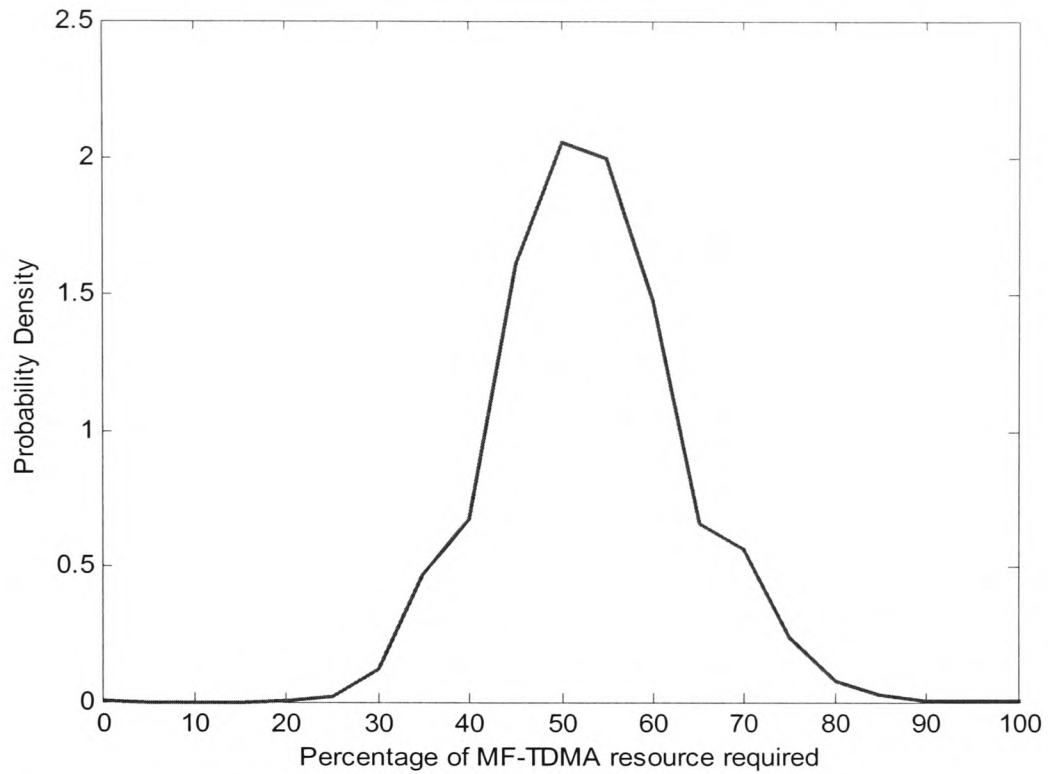


Figure 9.6: PDF of MF-TDMA resource required for all stations with traffic to maintain QoS parameters

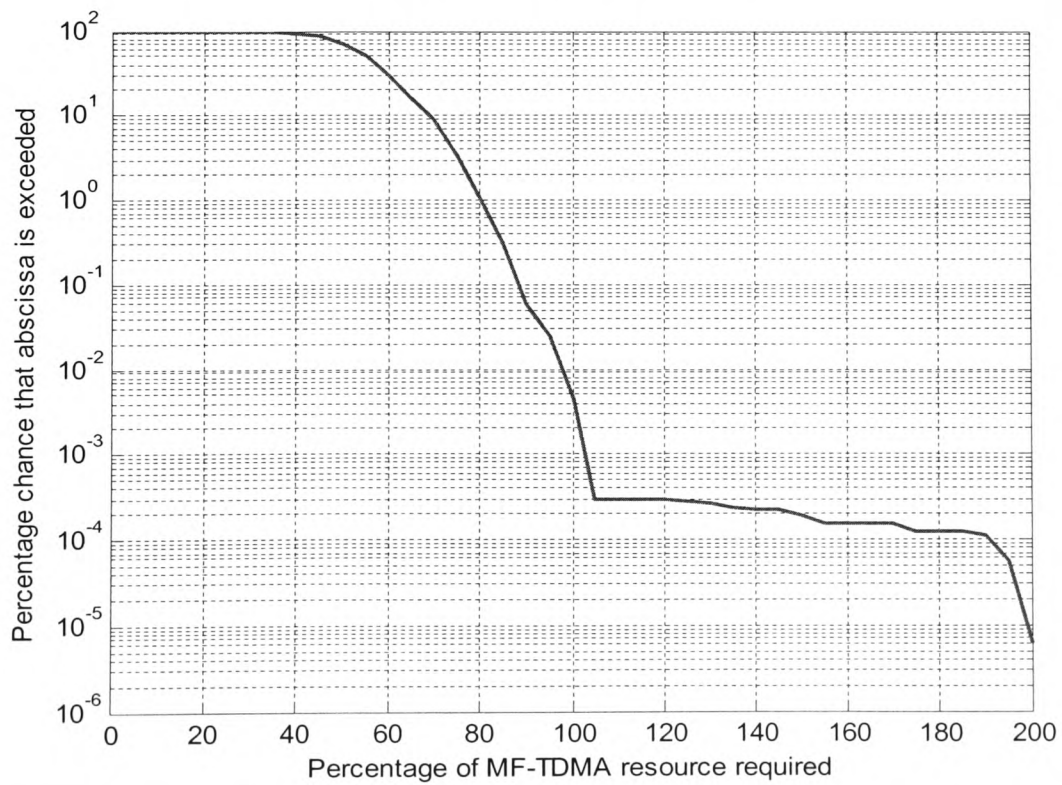


Figure 9.7: CDF of MF-TDMA resource required for all stations with traffic to maintain QoS parameters

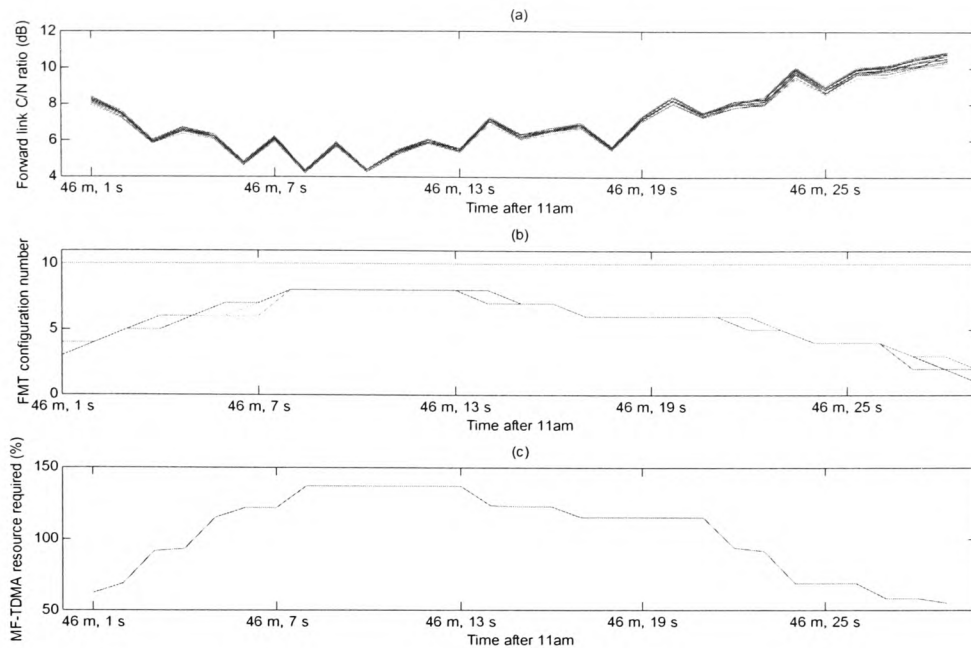


Figure 9.8: An example event from the 8th August 1998 showing a) the C/N ratio on the forward V band link, b) the FMT configuration number of the stations during the event and c) the percentage of MF-TDMA resource required for all stations to operate at the specified QoS

Figure 9.8a shows the C/N ratio for each of the 31 active stations in the system. Since all the C/N ratios are almost identical, it would appear that the fade is caused at the hub side, with each of the VSATs in clear air conditions. Figure 9.8b shows the FMT configuration for each of the active stations (see table 9.1 for the FMT parameters associated with each configuration). Finally figure 9.8c shows the percentage of the MF-TDMA resource that would be required for all stations to be accommodated in the frame. Obviously, not more than 100% resources could be allocated in reality and one or more stations would be denied resources at this point.

Figure 9.9 shows the same event for a single station. Figure 9.9a shows the C/N ratio and the C/N ratio required to communicate at the specified QoS. The C/N ratio never crosses the requirement threshold and it is clear that while the FMT reacts rapidly to the onset of fading, it reacts less quickly as the fading conditions return to normal. This is

due to the four second switching condition discussed in chapter 6. Figure 9.9b shows the FMT configuration of the station and figure 9.9c shows that there are no errors on the traffic channel for the entire duration of the event for this station.

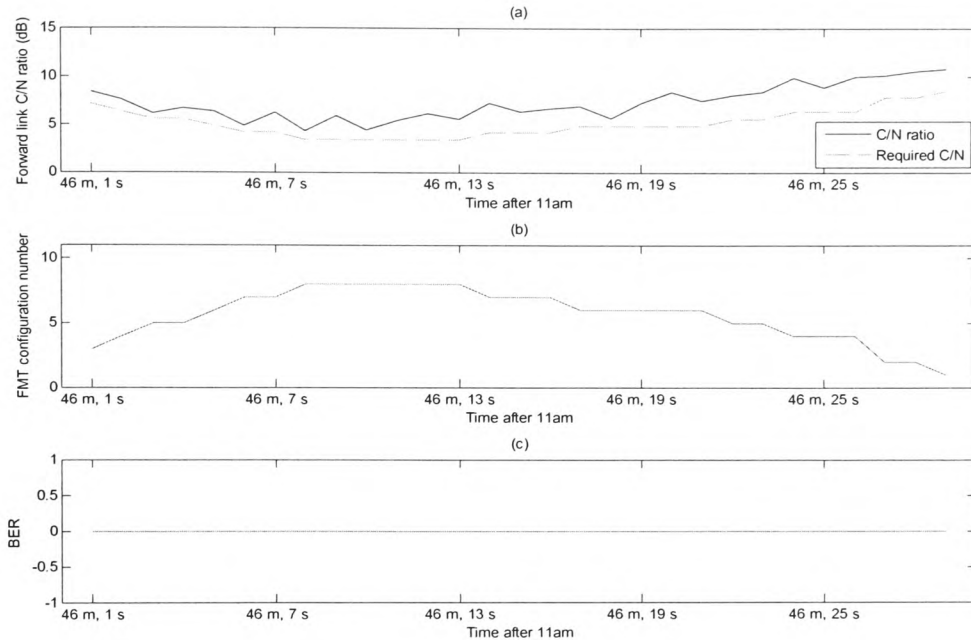


Figure 9.9: An example event for station 182 from the 8th August 1998 showing a) the C/N ratio and the C/N ratio required to meet QoS requirements, b) the FMT configuration number and c) the BER every second

While hub fading events are rare due to the site diversity employed, events at the VSAT are more common, an example of which is shown in figure 9.10. For this period, 52 VSATs are active, 48 of which occupy the *V* band MF-TDMA frame. This particular period of the 23rd of December 1997 was chosen as it reflects the situation where several stations experience separate fading events concurrently.

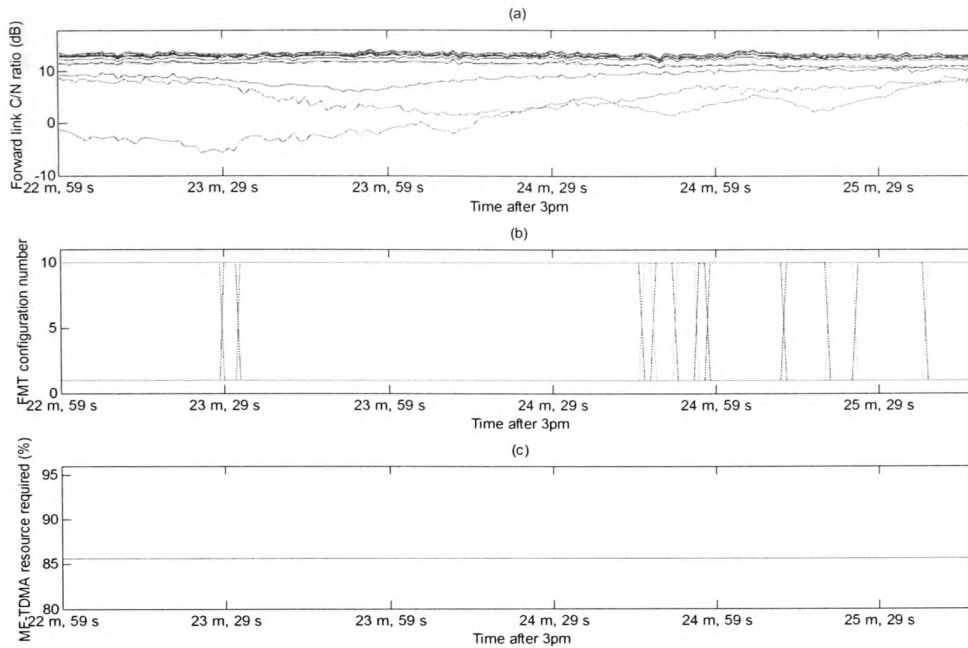


Figure 9.10: An example event from the 23rd December 1997 showing a) the C/N ratio on the forward *V* band link, b) the FMT configuration number of the stations during the event and c) the percentage of MF-TDMA resource required for all stations to operate at the specified QoS

Figure 9.10a shows the C/N ratios for each active station on the forward link. From this graph it is apparent that there are three stations within this period that experience fading independently. However in figure 9.10b, which shows the FMT configuration of all the stations, it is clear that no MF-TDMA resources are utilised as the only changes that occur are between the nominal *V* band scheme and the *Ku* band diversity channel. This is the same for most events and it was in fact very difficult to identify a period in which the signal processing method was utilised at all. Figure 9.10c merely confirms this observation since no change in MF-TDMA resource usage occurs.

Figure 9.11 shows this event for station number 391 merely to confirm that the QoS requirement is met for one of the fading stations.

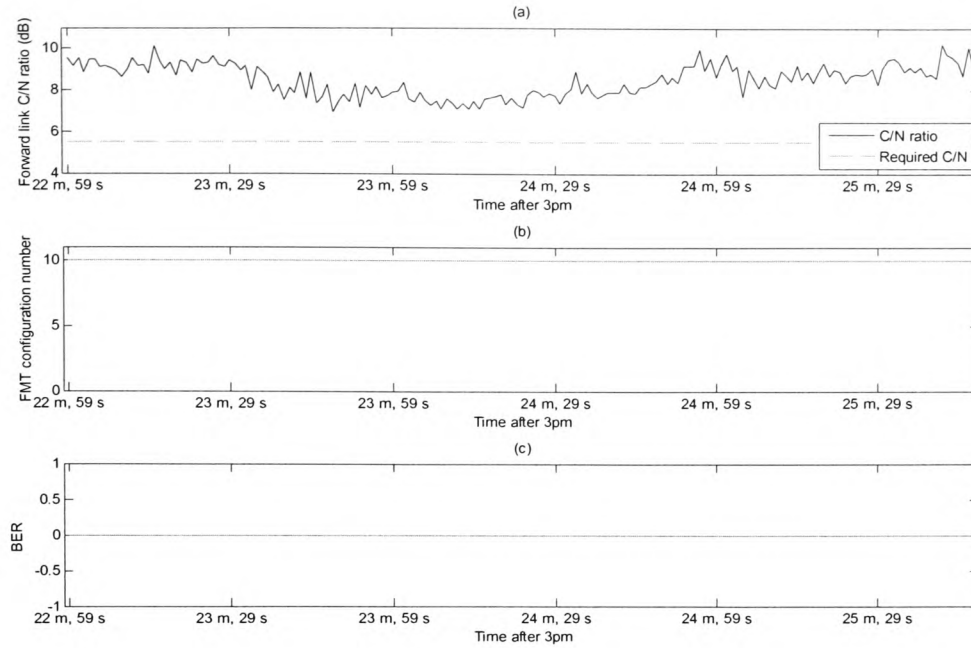


Figure 9.11: An example event for station 391 from the 23rd December 1997 showing a) the C/N ratio and the C/N ratio required to meet QoS requirements, b) the FMT configuration number and c) the BER every second

Also of interest is the proportion of resources that are devoted to fade mitigation in relation to traffic. This aspect has been consistently overlooked in the recent literature, most research neglecting it altogether, for which it must be assumed that FMT resources have been seen as simply having a small but unknown effect on the overall system performance. Figure 9.12 gives an indication of how much of the MF-TDMA resource is used for FMT purposes. If the proportion of FMT to normal traffic usage not exceeding the abscissa for 99.99% of the time is assumed to be a reliable figure to avoid having traffic resources ‘stolen’ by FMT usage, then in this case about 1% of the total MF-TDMA resource can be considered as utilised for FMT purposes. This is less than the 10% spare capacity that was provided for this scheme, as expected due to the addition of a traffic model.

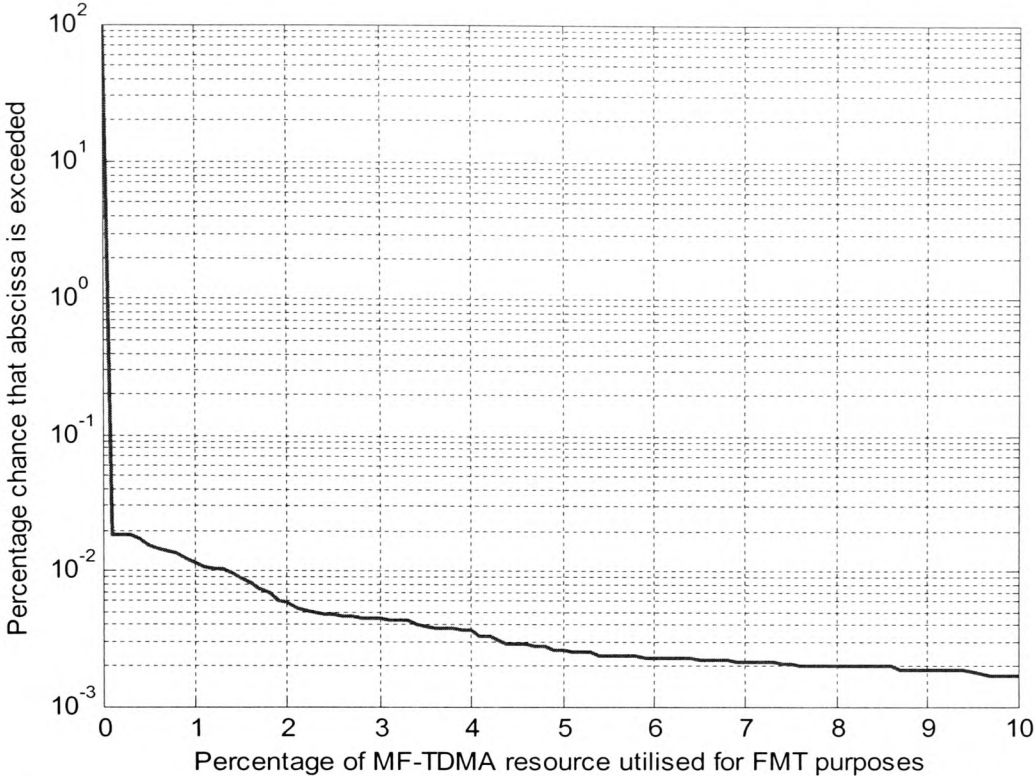


Figure 9.12: CDF of percentage of FMT contribution to overall MF-TDMA utilisation

9.3.3 Design and Results of a Single-station Frequency Diversity Approach

A second system was also simulated where a single station was accommodated in the *Ku* band to determine the performance difference related to the amount of resources allocated to frequency diversity and to the signal processing method. In this case, a maximum of 57 stations are allocated a burst in the MF-TDMA frame, instead of 54 as was the case in the previous section. One disadvantage of this is that the frame format between the *V* and *Ku* bands is now different. This may further complicate the design of the VSAT controllers, but allows a more flexible assignment of resources.

The FMT configurations remain the same as those presented in table 9.1, while the link budgets were recalculated based on the altered FMT scenario. The only exceptions to the configurations in table 9.1 are the burst durations, the updated versions of which now appear in table 9.4. The most relevant parameters for the link budgets appear in tables 9.5 and 9.6 below. It is assumed that the stations are located within the same region as stated previously.

Configuration Number	Freq. Band	Modulation	Code Rate	C/N Required (dB)	Forward / Return Link Burst Durations (ms)
1	<i>V</i>	QPSK	6/7	8.5	0.320 / 5.317
2	<i>V</i>	QPSK	4/5	7.9	0.342 / 5.660
3	<i>V</i>	QPSK	3/4	7.2	0.365 / 6.022
4	<i>V</i>	QPSK	2/3	6.7	0.409 / 6.727
5	<i>V</i>	QPSK	1/2	5.8	0.545 / 8.862
6	<i>V</i>	BPSK	4/5	4.9	0.683 / 11.32
7	<i>V</i>	BPSK	3/4	4.2	0.729 / 12.04
8	<i>V</i>	BPSK	2/3	3.7	0.819 / 13.45
9	<i>V</i>	BPSK	1/2	2.8	1.089 / 17.72
10	<i>Ku</i>	QPSK	1/2	5.8	20 / 20

Table 9.4: FMT configurations

Hub antenna diameter	2 m
Hub antenna efficiency	68%
VSAT antenna diameter	0.8 m
VSAT antenna efficiency	60%
Hub power (Ku band link)	200 W
Hub power (V band link)	5 W
Satellite G/T at Ku band	8 dBi/K
Satellite G/T at V band T_s of 900K	16.7 dBi/K
Satellite system noise temperature at Ku band	500 K
Earth station system noise temperature at Ku band	150 K
Satellite system noise temperature at V band	900 K
Earth station system noise temperature at V band	350 K
Satellite EIRP at Ku band	51 dBW
Satellite EIRP at V band	61.8 dBW
Number of traffic carriers at Ku band	1
Number of traffic carriers at V band	1
Number of control channel carriers at Ku band	1
Number of stations at Ku band	1
Maximum number of stations at V band	57
Burst rate of each Ku band traffic channel (QPSK modulation, 1/2 rate coding, $5\mu s$ guard time)	1.145 Mbps
Burst rate of each V band traffic channel (QPSK modulation, 6/7 rate coding, $5\mu s$ guard time)	42.27 Mbps
Burst rate of control channel (BPSK modulation, 1/2 rate coding, $5\mu s$ guard time)	110.5 kbps
Occupied Ku band transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	772.9 kHz
Occupied V band transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	28.53 MHz
Occupied Ku band transponder bandwidth for control channel (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	149.1 kHz
Minimum C/N for BER of 10^{-6} at Ku band	5.51 dB
Minimum C/N for BER of 10^{-6} at V band	8.51 dB
Minimum C/N for BER of 10^{-6} for control channel (turbo code)	2.5 dB
Total information bit rate available for 58 stations at 512 kbps	29.7 Mbps
Total information bit rate including ATM header overhead	33.2 Mbps
Ku -band uplink frequency	13.87 GHz
Ku -band downlink frequency	11.35 GHz
V band uplink frequency	50.42 GHz
V band downlink frequency	39.52 GHz
Miscellaneous losses	0.5 dB
Gaseous losses at Ku band uplink frequency	0.7 dB
Gaseous losses at Ku band downlink frequency	0.5 dB
Gaseous losses at V band uplink frequency	3 dB
Gaseous losses at V band downlink frequency	1 dB

Table 9.5: Most significant link budget parameters for the forward link with one station at Ku band

Hub antenna diameter	2 m
Hub antenna efficiency	68%
VSAT antenna diameter	0.8 m
VSAT antenna efficiency	60%
VSAT power (<i>Ku</i> band link)	2 W
VSAT power (<i>V</i> band link)	0.8 W
Satellite G/T at <i>Ku</i> band T_s of 500K	8 dBi/K
Satellite G/T at <i>V</i> band T_s of 900K	16.7 dBi/K
Satellite system noise temperature at <i>Ku</i> band	500 K
Earth station system noise temperature at <i>Ku</i> band	150 K
Satellite system noise temperature at <i>V</i> band	900 K
Earth station system noise temperature at <i>V</i> band	350 K
Satellite EIRP at <i>Ku</i> band	51 dBW
Satellite EIRP at <i>V</i> band	61.8 dBW
Number of traffic carriers at <i>Ku</i> band	1
Number of traffic carriers at <i>V</i> band	16
Number of control channel carriers at <i>Ku</i> band	1
Number of stations at <i>Ku</i> band	1
Maximum number of stations at <i>V</i> band	57
Burst rate of each <i>Ku</i> band traffic channel (QPSK modulation, 1/2 rate coding, 5 μ s guard time)	169.6 kbps
Burst rate of each <i>V</i> band traffic channel (QPSK modulation, 6/7 rate coding, 5 μ s guard time)	397.2 kbps
Burst rate of control channel (QPSK modulation, 1/2 rate coding, 5 μ s guard time)	94.84 kbps
Occupied <i>Ku</i> band transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	498.4 kHz
Occupied <i>V</i> band transponder bandwidth (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	4.930 MHz
Occupied <i>Ku</i> band transponder bandwidth for control channel (36 MHz total, roll-off factor of 0.35, 40 kHz guard band)	64.02 kHz
Minimum C/N for BER of 10^{-6} at <i>Ku</i> band	5.51 dB
Minimum C/N for BER of 10^{-6} at <i>V</i> band	8.51 dB
Minimum C/N for BER of 10^{-6} for control channel (convolutional code)	7.5 dB
Total information bit rate available for 58 stations at 64 kbps	3.71 Mbps
Total information bit rate including ATM header overhead	4.92 Mbps
<i>Ku</i> -band uplink frequency	13.87 GHz
<i>Ku</i> -band downlink frequency	11.35 GHz
<i>V</i> band uplink frequency	50.42 GHz
<i>V</i> band downlink frequency	39.52 GHz
Miscellaneous losses	0.5 dB
Gaseous losses at <i>Ku</i> band uplink frequency	0.7 dB
Gaseous losses at <i>Ku</i> band downlink frequency	0.5 dB
Gaseous losses at <i>V</i> band uplink frequency	3 dB
Gaseous losses at <i>V</i> band downlink frequency	1 dB

Table 9.6: Most significant link budget parameters for the return link with one station at *Ku* band

The same statistical analysis was carried out in this scenario as for the previous one. Figure 9.13 shows the annual availability of each of the 620 stations. There is little visible difference from the previous scenario. Figure 9.14 shows that the variance of the FMT method remains far superior to the fixed margin method, despite the reduction in *Ku* band resources.

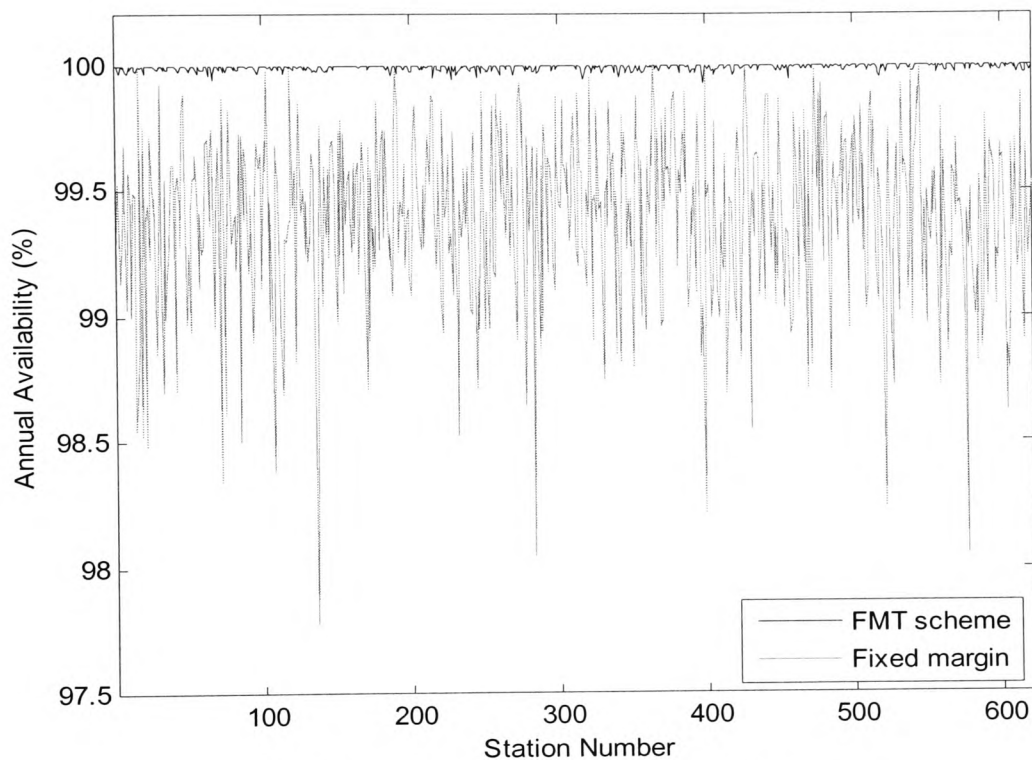


Figure 9.13: Annual availability of stations

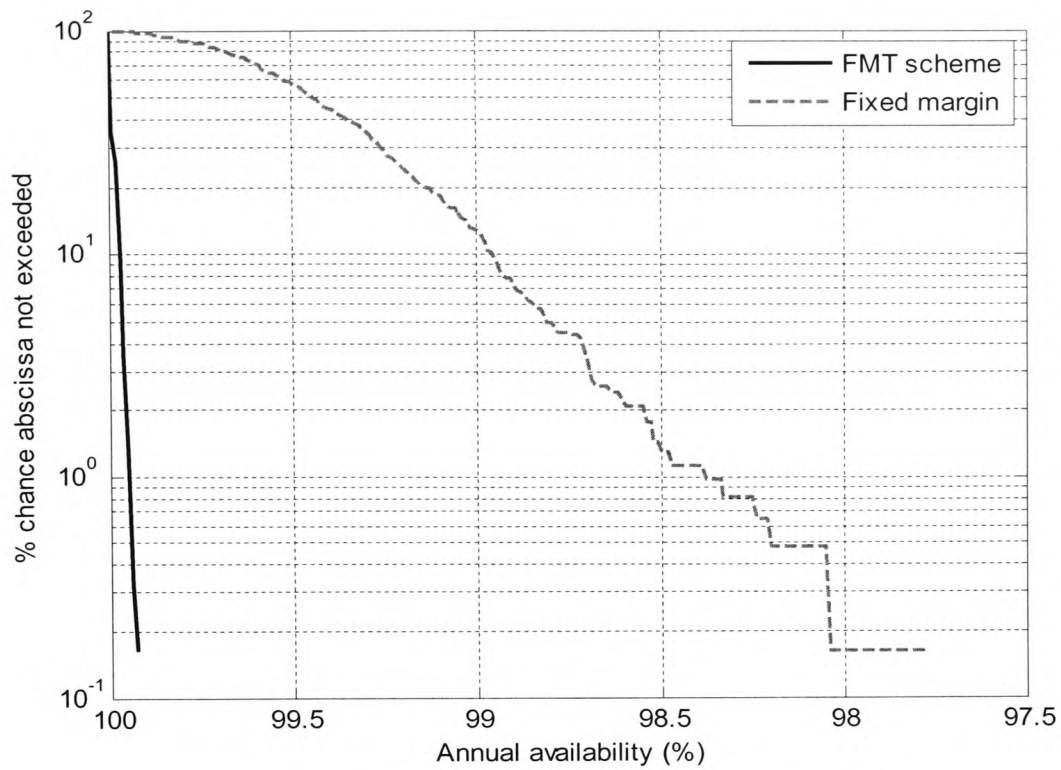


Figure 9.14: CDF of annual availability of stations

The PDF and CDF of the MF-TDMA resource requirement are shown in figures 9.15 and 9.16 respectively. There is little meaningful difference between these and the previous results for the MF-TDMA resource requirement with 4 *Ku* band stations, the chance of frame exhaustion being lower than 0.001% in both cases.

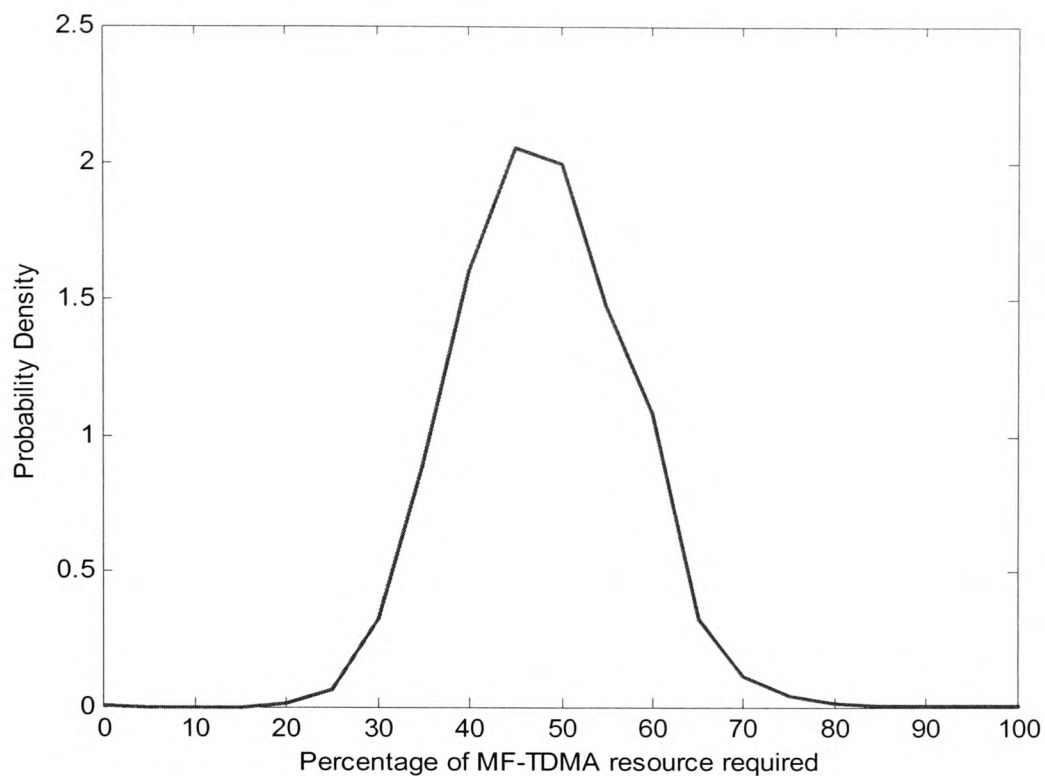


Figure 9.15: PDF of MF-TDMA resource required for all stations with traffic to maintain QoS parameters

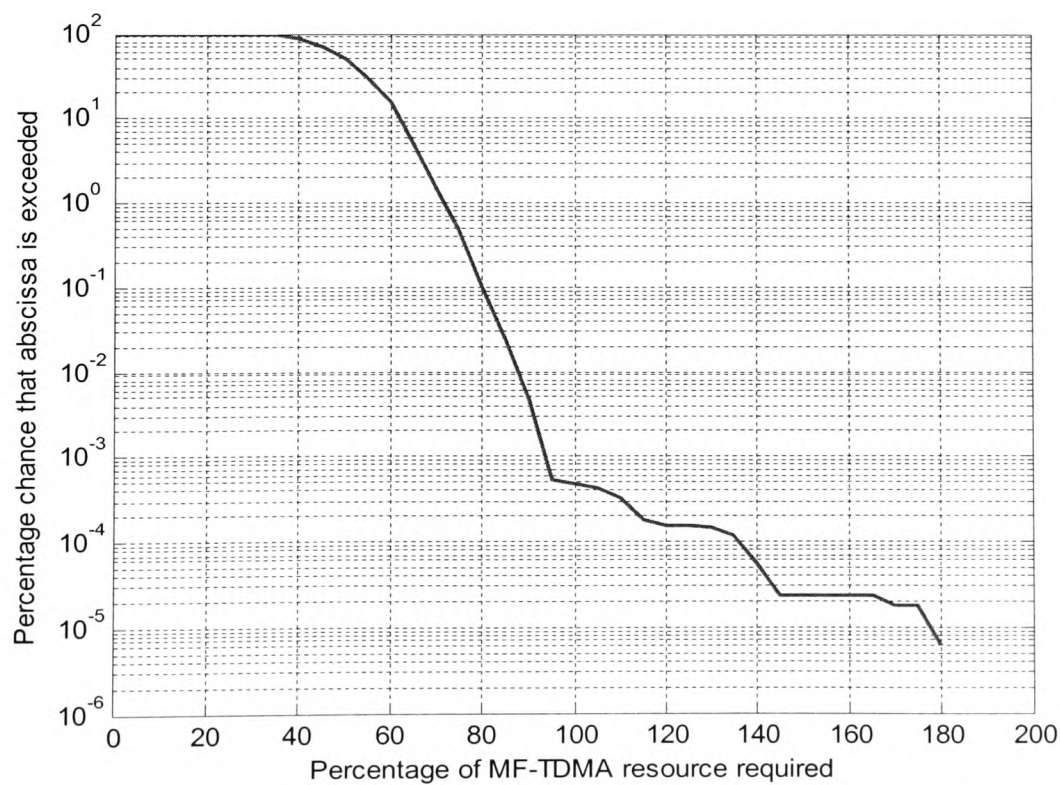


Figure 9.16: CDF of MF-TDMA resource required for all stations with traffic to maintain QoS parameters

The hub fading event that was shown in figures 9.8 and 9.9 was repeated for the scheme with 1 *Ku* band station. Figures 9.17 and 9.18 show the equivalent of figures 9.8 and 9.9 respectively. Again, there is little meaningful difference apart from a slight rise in the maximum MF-TDMA frame requirement for the scenario with a single *Ku* band station, which would be expected since more stations now make use of the signal processing method. However, frame exhaustion occurs in both scenarios, showing that the extra *Ku* band stations have little effect in the case of hub-fading.

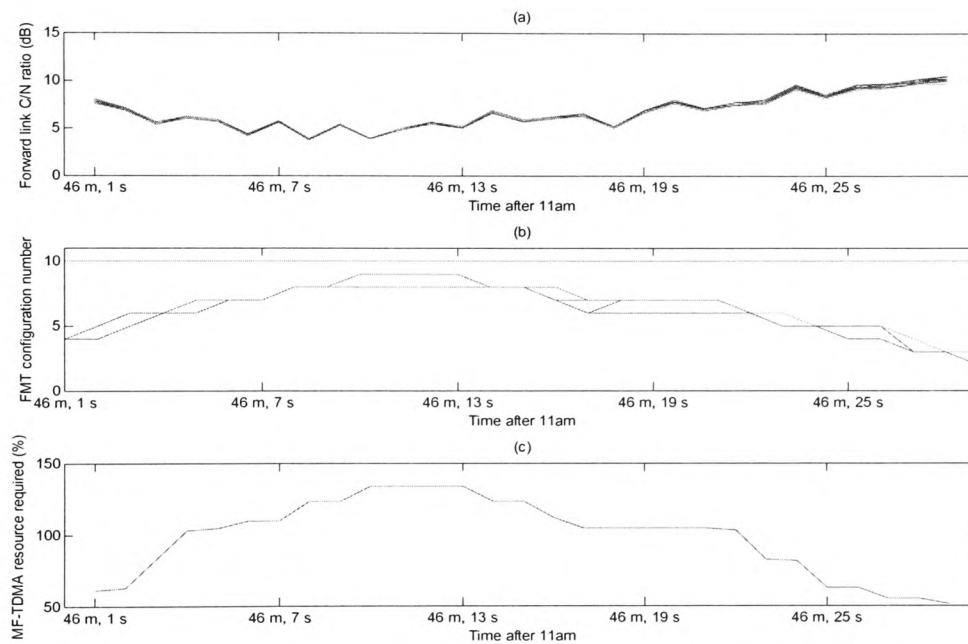


Figure 9.17: An example event from the 8th August 1998 showing a) the C/N ratio on the forward *V* band link, b) the FMT configuration number of the stations during the event and c) the percentage of MF-TDMA resource required for all stations to operate at the specified QoS

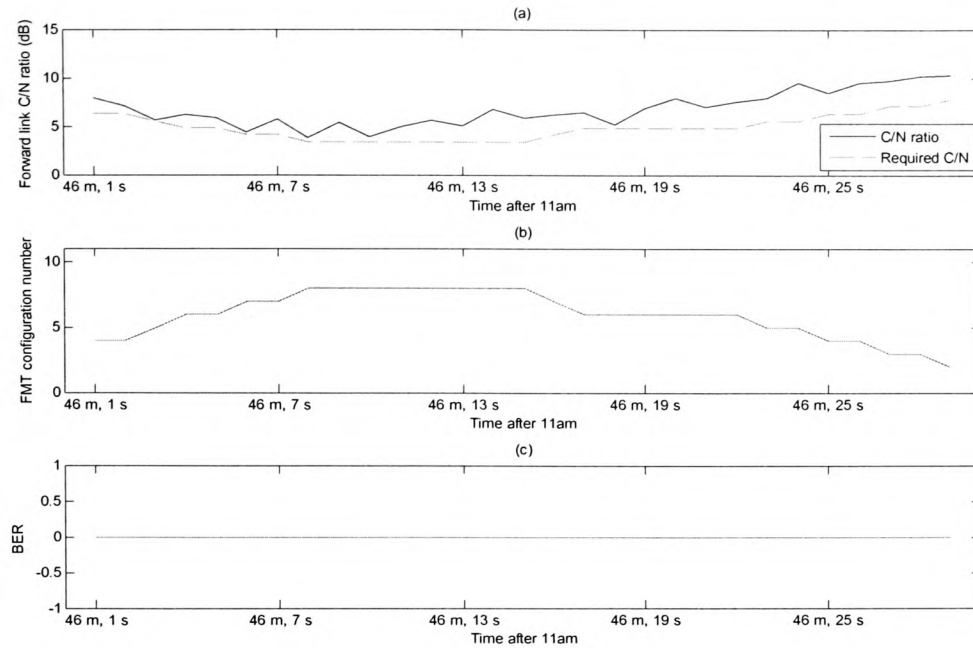


Figure 9.18: An example event for station 182 from the 8th August 1998 showing a) the C/N ratio and the C/N ratio required to meet QoS requirements, b) the FMT configuration number and c) the BER every second

Figures 9.19 and 9.20 repeat the results for the same VSAT fading period as shown in the previous scenario. It is clear from figure 9.19b that the MF-TDMA resources are more highly utilized in the case with a single *Ku* band station. However, figure 9.20 shows that the QoS for the station in question is unaffected despite the change in FMT type.

The effect on MF-TDMA resource usage of a reduction in frequency diversity provision can be seen in figure 9.21, which shows how much of the resource is dedicated to FMT usage. The percentage of FMT resources out of the total utilised resources not exceeded for 99.99% is now 2%, double what it was previously. The graph drops sharply from 0%, where most samples occur since it is only raining for a small proportion of the time. However, beyond the initial drop, the fall in this case is not as rapid as for the scenario with 4 *Ku* band stations since more MF-TDMA resources are now utilized.

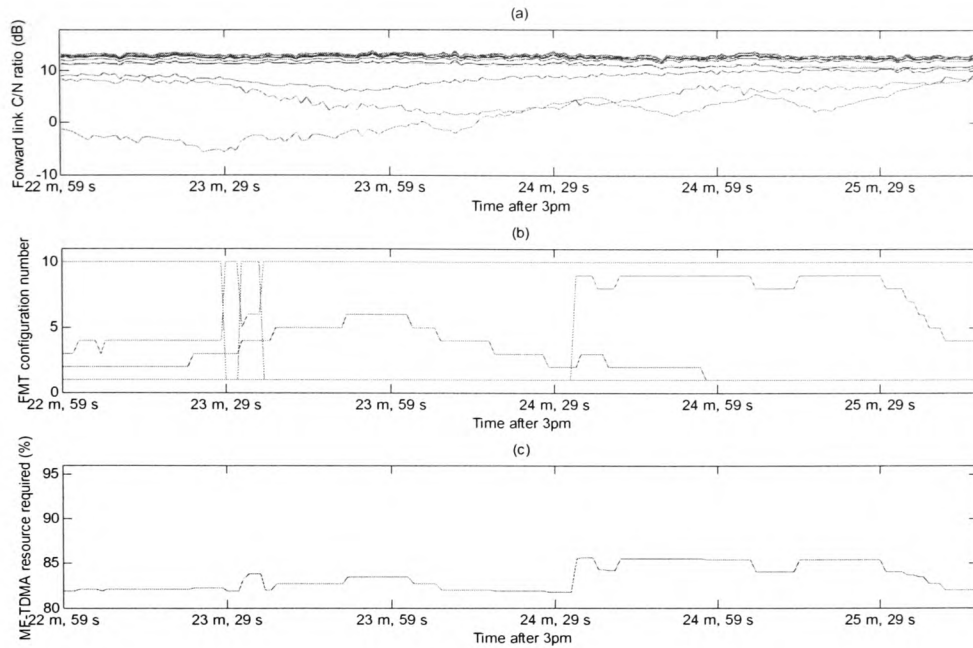


Figure 9.19: An example event from the 23rd December 1997 showing a) the C/N ratio on the forward V band link, b) the FMT configuration number of the stations during the event and c) the percentage of MF-TDMA resource required for all stations to operate at the specified QoS

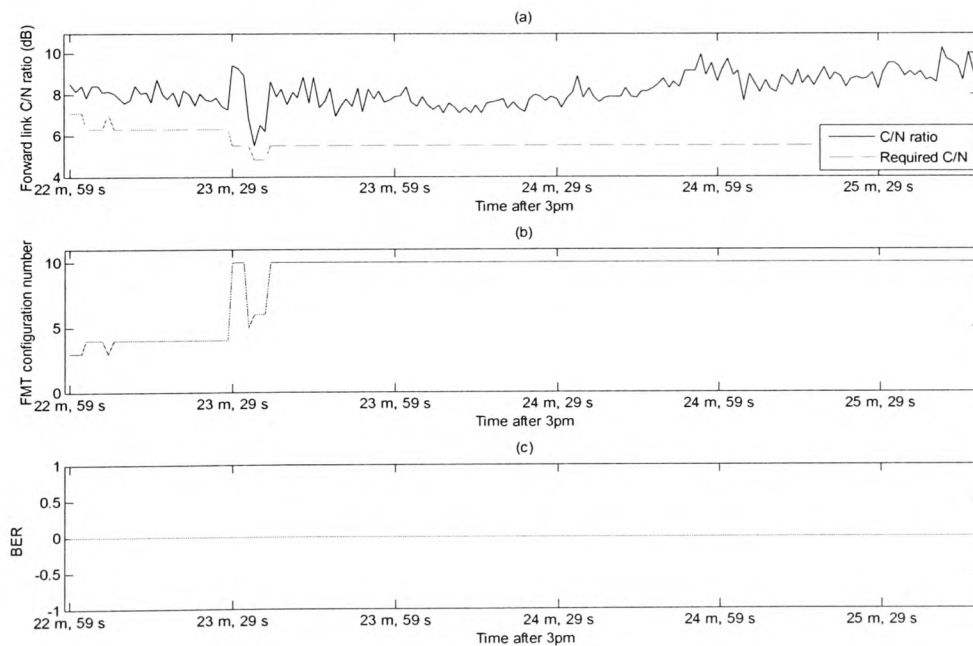


Figure 9.20: An example event for station 391 from the 23rd December 1997 showing a) the C/N ratio and the C/N ratio required to meet QoS requirements, b) the FMT configuration number and c) the BER every second

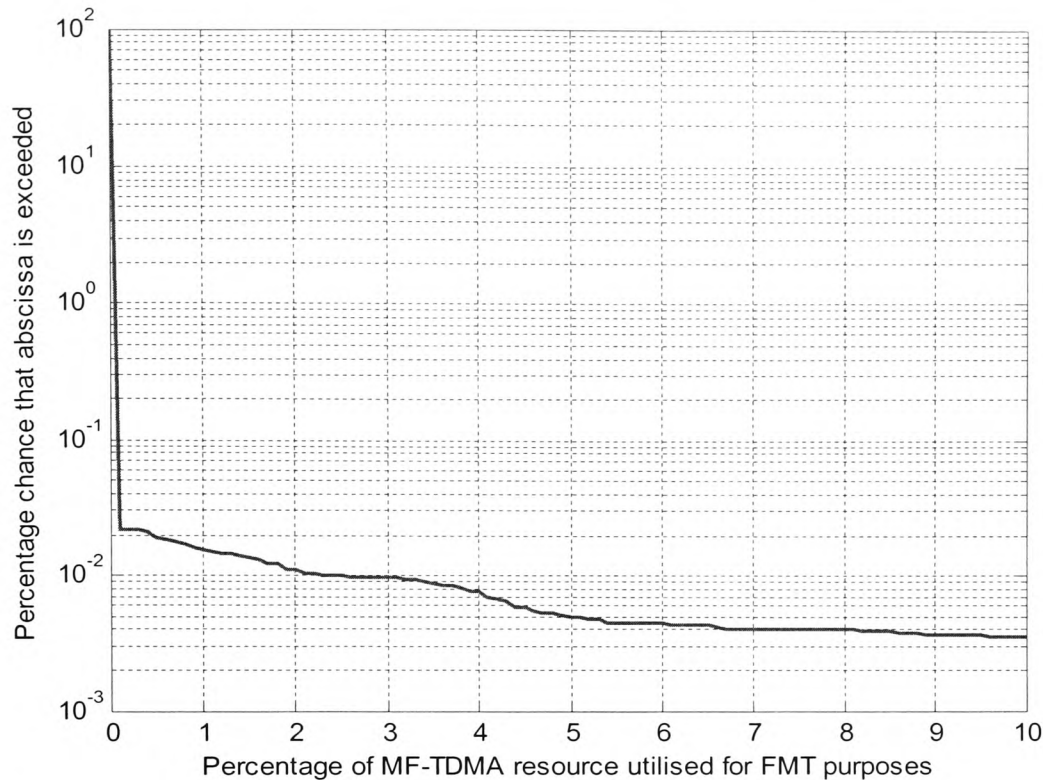


Figure 9.21: CDF of percentage of FMT contribution to overall MF-TDMA utilisation

Finally, figure 9.22 shows the robustness of the control channel against fades in comparison to the traffic channel. Figure 9.22a shows the C/N ratio on the forward link and that required to meet QoS requirements for station 534 in the same event from the 23rd December 1997 as seen previously. The dashed line representing the required C/N ratio is not plotted in regions where the station is denied resources but the errors that would occur should the station remain active are resolved nonetheless. In figure 9.22b the forward control channel is shown to be without error for the entire duration, whereas in figure 9.22c the traffic channel is shown to have errors when the dynamic range of the FMT is exceeded. In other cases, rain fading could also be sufficient to cause errors on the control channel, although by the time this were to occur, the station would already have been denied resources until such a time as the traffic channel conditions returned to a state in which communication could be achieved at the desired BER. At

this point, the control channel, being far more robust than the traffic channel, would also be able to communicate successfully. This functionality is essential to the successful operation of the protocol since control channel errors affect every station, not just the station with fading since a single error could cause a station to transmit at the wrong instant resulting in collision with another station in the MF-TDMA frame.

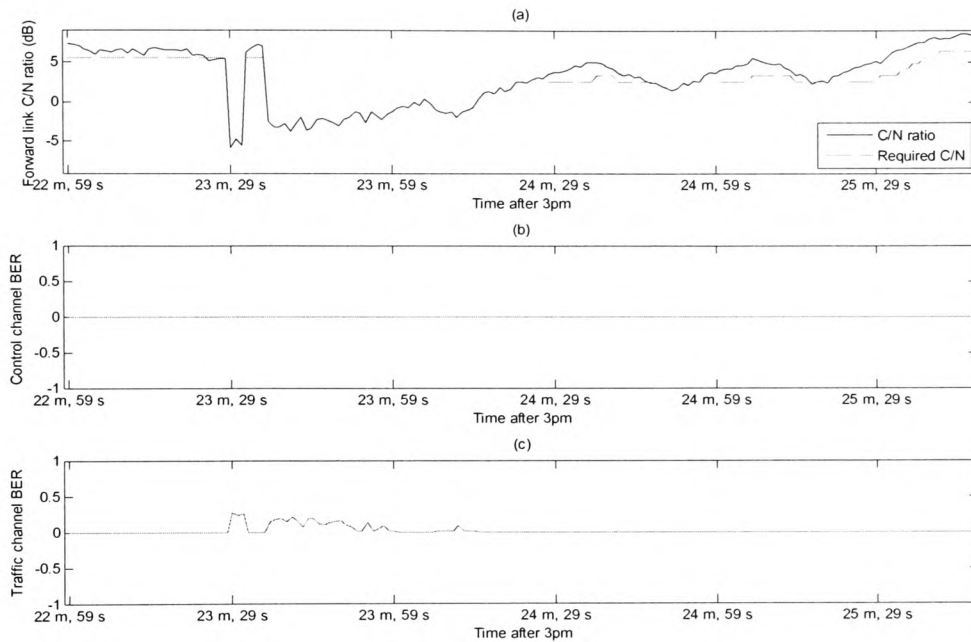


Figure 9.22: An example event for station 534 from the 23rd December 1997 showing a) the C/N ratio and the required C/N ratio to meet QoS requirements on the forward traffic channel, b) the BER on the forward control channel and c) the BER on the forward traffic channel

As already mentioned, information on the amount of resources required for FMT purposes is lacking from the recent literature. Authors are either more interested in the effect of different traffic models on the resource usage, such as in [139] or the effects of different classes of traffic, as in [140] and [141] or in the difference between resource allocation strategies such as in [72] and [142]. Although the exact amounts required for FMT resources depend upon the FMT schemes employed, in this particular case the following observations can be made:

Firstly, when dimensioning the system either of two approaches can be taken. The first involves assuming a certain portion of the MF-TDMA frame(s) will be utilized for FMT purposes and designing these frames with some spare capacity as has been done in this work, although 10% extra has been shown to be a significant over-compensation. The second is to design the frame for traffic only and adjust the number of stations in the system to bring the statistics in line with the extra resources that will be required for FMT purposes. Either of these methods is an improvement from the current situation where FMT resources are assumed to have a negligible effect on the overall system.

Secondly, when using a hybrid FMT solution as has been explored in this work, the proportion of Ku band resources must be carefully dimensioned so as not to make the signal processing method redundant and waste what will amount to a more sought after and therefore expensive resource. It is likely that the percentage of resources allocated at Ku band should be in the region of the likelihood that rain attenuation will prohibit any station from communicating at V band, since this is when the Ku band resource is most useful. The forward downlink effective margin is about 7 dB which is exceeded by rain less than 1% of the time. The scenario which accommodates 4 stations at Ku band is a total of 6.9% of the full capacity, which is the reason that very little use of the signal processing technique can be seen. The scenario with just a single Ku band station has a Ku band capacity of just 1.7% of the full capacity which is closer to the V band outage threshold probability. If there were even more stations in the system, perhaps spread across several transponders, the Ku band resource could be optimized even further.

9.4 *Conclusions*

In summary, this chapter contains several significant findings. Firstly, it has been shown that a hybrid FMT that combines site diversity and power control to reduce the impact of hub fades and frequency diversity and signal processing to reduce the impact of VSAT fades is capable of increasing availability from the fixed link budget scenario. This is a significant step towards affordable V band exploitation, with roughly 46 minutes of outage in a year and a small amount of Ku band utilisation.

Secondly, use of the hybrid FMT, while increasing overall availability, also reduces station-to-station outage variability, giving further assurances for QoS guarantees.

Finally, the amount of resources required by the FMTs in comparison with the resources required by the traffic itself has been compared, providing a first estimate of how a working system could be dimensioned to allow for the extra overhead of fade mitigation.

Chapter 10

Conclusions and Further Work

10.1 Conclusions

This thesis has investigated several elements of an FMT system applicable to satellite internet provision at V band. A summary of these elements is shown below:

- A review of the various FMTs available with an analysis that was specific to satellite internet delivery in chapter 2.
- A review of fade dynamics at V band in chapter 3.
- Implementation of a suitable turbo-coding scheme in chapter 4.
- An approximation of the effects of site diversity suitable for a single time series of attenuation in chapter 5.
- A review of short-term fade prediction methods featuring an ADALINE predictor design and a hybrid of two signal processing predictors in chapter 6.
- A description of the pre-processing that was required, the traffic assumptions that were made and a dual channel protocol in chapter 7.
- A preliminary investigation of three FMT scenarios in chapter 8.
- A final and more detailed investigation of the hybrid FMT scenario in chapter 9.

Many conclusions may be drawn from all stages of this work, some of which can be drawn from individual chapters whereas others rely on the information from several chapters. An effort has been made to order the conclusions in a format relevant to the chapter layout of the thesis.

Fade mitigation techniques and fade mitigation control has been reviewed for the specific purpose of satellite internet delivery. Several complementary schemes were

Chapter 10 – Conclusions and Further Work

chosen, including adaptive modulation, adaptive coding, adaptive MF-TDMA and frequency diversity to compensate for VSAT fading, and uplink power control and site diversity to account for hub fading. However, these are not the only schemes available; other available schemes were data rate reduction by adaptive fade spreading, adaptive CDMA and orbital diversity, however, since the combined gains of the other methods were deemed sufficient, these methods were not investigated.

Several schemes were also ruled out altogether including direct bit-rate reduction, which reduced downlink power output and was of little use, and downlink power control and on-board beam shaping which both require processing on-board the satellite and thereby contravene the assumption throughout this work that a bent-pipe transponder can be utilised.

A review of second order fade dynamics at V band, including fade slope, fade duration and inter-fade duration has been carried out. The current ITU-R model for fade slope was found to give a good fit visually although it did not pass a chi-squared goodness of fit test [18]. No significant diurnal variation was found for the data from RAL, although a seasonal variation was apparent, with autumn showing the highest standard deviation. The ITU-R model for fade duration was not found to fit the measurements particularly well visually and no chi-squared test was required to conclude that the model is unreliable. No diurnal or seasonal variation was found.

In light of the kind of fade mitigation controller employed, where real-time fading conditions are passed between VSAT and hub, the statistical evaluation of fade dynamics was eventually not found to be useful. Before creating the link budgets, it was

not obvious that the threshold levels would be fixed so definitely by the coding scheme employed as the original idea had been to set different threshold levels depending on the fade slope statistics. Since this was ultimately not possible, fade dynamics statistics played no direct role in the final simulations.

A DVB-RCS turbo code has been implemented. This was chosen for the purpose of satellite internet delivery for its high coding gain and for its suitability in conjunction with ATM sized frames, which were also chosen for use within the protocol. The main conclusions that were drawn from this work were the values of E_b/N_0 that were required for communication at a specified BER (10^{-6} in this case). However, also of note during this stage was the failure of the Matlab software to process its native m-files at sufficient speed to simulate millions of bits to generate the necessary graphs. For this reason, many of the functions that required a lot of processing were written or rewritten as C mex files that could be compiled and executed in the same way as Matlab m-files but executed between 20 and 200 times faster than Matlab's interpreted language. If it were not for this functionality, the final simulations would not have been possible.

A method of reproducing the effect of site diversity using a single time series of attenuation has been developed. This was achieved firstly by an analysis of rain rate at two sites close to that of the RAL satellite beacon measurement and secondly by correlating the concurrency of rain with two points separated in time for the time series of attenuation. A value of 14.5 minutes was found to correlate most accurately with the rain statistics and this was then tested against the ITU-R model for site diversity improvement, which showed a good visual fit. Finally it was concluded that the spatial separation of site diversity could be approximated by a separation in time with a single

time series of attenuation, although no attempt was made to expand this process into a model for a general case.

Short-term fade predictors from the literature have been compared with ADALINE and hybrid signal processing predictors in terms of their performance for a set of rain events within the time series of attenuation collected by RAL. Perhaps the most important of the conclusions from this chapter was that the predictors that are based on rain fade slope should be avoided as they perform poorly in comparison with the signal processing and statistical methods. Of the other short-term fade predictors under comparison none outperformed another to such an extent that a definite favourite can be identified.

What could be concluded however, is that although in terms of performance the hybrid predictor had a slight advantage over the ADALINE, ARMA and statistical methods, it also required the most processing, closely followed by the ARMA predictor. Therefore, if processing is not a concern, the hybrid should be chosen over the ARMA predictor. The ADALINE and statistical methods both required little processing but neither performed quite as well as the hybrid predictor. Of these two methods, neither could be clearly labelled as superior in terms of performance but the one advantage the statistical method had over the ADALINE was that it was not limited to predicting attenuation with a 50% exceedance probability but was able to predict for any exceedance probability. This functionality was not required in the controller used in this thesis as the ADALINE method was combined with a fade mitigation control method described at the end of chapter 6.

A protocol structure has been designed in order to create link budgets. These showed very clearly that a frequency diversity system was crucial to the implementation of any high availability link at V band. However, this conclusion consequently removes the need for a dual channel protocol. This was initially useful for designing different availabilities for the control and traffic information at V band but would not be necessary in a frequency diversity scheme since the station or stations using the lower frequency channel are sufficiently protected for a high availability link anyway. Since this channel adds significant complexity to the potential hardware controllers, a better option may be to reintroduce the control information into a burst in the traffic channel, and maintain only a single channel in each direction, such as that used in the protocol in [13].

Three simulations were carried out for signal processing, frequency diversity and hybrid FMT scenarios. This work reinforced the conclusion that frequency diversity is a requirement for a high availability system as the simulation relying on only V band signal processing techniques was shown to be inferior to the frequency diversity and hybrid FMT schemes. Another useful conclusion that was drawn from this work was the complementary nature of the frequency diversity and signal processing methods. The frequency diversity method was able to provide a large dynamic range to a small number of stations while the signal processing method was able to provide a smaller dynamic range to a greater number of stations. We could therefore conclude that while frequency diversity was necessary for a high availability V band communications system, signal processing was also an important addition. In addition, the effect of site diversity was also investigated, and this led to the conclusion that site diversity was also invaluable to a satellite internet communications system. In fact the role played by site

Chapter 10 – Conclusions and Further Work

diversity in mitigating hub attenuation was very similar to that played by frequency diversity in mitigating VSAT attenuation. These conclusions led to a final set of simulations with a hybrid FMT.

A hybrid FMT scenario for different mixes of frequency diversity and signal processing techniques has been simulated. It was shown, given the traffic assumptions of chapter 7, that a hybrid FMT scheme with a single station at *Ku* band and 2% of the total resources dedicated to FMT usage was the optimum. The percentage of *Ku* band links required of the total was related to the outage probability of a link and therefore the number of *Ku* band stations would increase as the number of overall users increased. However, this could also be achieved by dimensioning the system with less stations rather than adding capacity specifically for FMT use.

This scenario was able to provide high availability communication at *V* band with just 46 minutes of annual outage. This is comparable with the availability of current fixed margin systems at *C* band and shows that successful exploitation of *V* band frequencies is technically possible. Although it remains to be seen whether this system is economically viable, this work is a significant step towards satisfying a growing demand for satellite bandwidth.

10.2 Further Work

There are several directions in which this work may be continued. The first would consist of expanding the simulation into the upper layers of the OSI network model. This would require a large amount of work in simulating a network architecture in Matlab and may not ultimately be a worthwhile endeavour since simpler software packages are already available for the network layer such as NS-2 and OPNET. Also, the simulations already carried out in Matlab for chapter 9 took several days each to complete with the hardware currently available. A second option to continue the work would be to build a hardware controller and test the protocol, or a single channel version of the protocol with a real satellite link. This option may be better than the first since a hardware scenario would resolve any potential problems with the protocol. A final option would be to repeat the simulations of chapter 9 both to fine-tune the link budgets, and to compare the system with different traffic models, different resource allocation strategies or by comparing the performance with the VSATs correlated by using a similar method to that used for creating site diversity.

References

References

- [1] P. S. Yeung, "INTELSAT's broadband strategy," presented at IEE Seminar on Broadband Satellite: The Critical Success Factors Technology, Services and Markets, London, UK, 2000.
- [2] M. Sooriyabandara and G. Fairhurst, "Implications of broadband satellite BoD algorithms on TCP performance," presented at IEE Seminar Personal Broadband Satellite, London, UK, 2002.
- [3] A. Fidler, T. Pell, D. Bryant, I. Rose, I. Moffat, and A. Grace, "BTopenworld satellite broadband from concept to launch," presented at IEE Seminar Personal Broadband Satellite, London, 2002.
- [4] Tooway - The new way to broadband, <http://www.tooway.net>, accessed on 30th November 2007
- [5] S. Ventouras, C. L. Wrench, and S. A. Callaghan, "Earth-space propagation - Measurement and analysis of satellite beacon transmissions at frequencies up to 50 GHz," Project report for the Radio Communications Agency, Didcot, UK 2000.
- [6] "Attenuation by atmospheric gases," in *Recommendations of the ITU-R*, vol. P.676-5, 2001.
- [7] A. Savvaris, "Accurate prediction of scintillation degradation applicable to satellite communication systems design," Ph.D. Thesis, School of Electronics, University of Glamorgan, 2004
- [8] K. Al-Ansari, P. Garcia, J. M. Riera, and A. Benarroch, "One year cloud attenuation results at 50 GHz," *Electronics letters*, vol. 39, no. 1, pp. 136-137, 2003.
- [9] "Propagation data and prediction methods required for the design of Earth-space telecommunication systems," in *Recommendations of the ITU-R*, vol. P.618-8, 2003.
- [10] T. Pratt, C. Bostian, and J. Allnut, *Satellite Communications*: Wiley, 2003.
- [11] "Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems," vol. ETSI EN 301 790 v1.41, 2005.
- [12] ISO/IEC, "Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model," 1994.
- [13] N. Celandroni, E. Ferro, N. James, and F. Potorti, "FODA/IBEA-TDMA: A flexible fade countermeasure system for integrated services in user-oriented networks," *International Journal of Satellite Communications*, vol. 10, pp. 309-323, 1992.
- [14] "HYREX: The Hydrological Radar Experiment Special issue," in *Hydrology and Earth System Sciences*, vol. 4, 2000, pp. 523-530.
- [15] "Specific attenuation model for rain for use in prediction methods," in *ITU-R Recommendation P.838-2*, 2003.
- [16] "Prediction method of fade dynamics on Earth-space paths," in *ITU-R*, vol. P.1623-1, 2005.
- [17] A. Chambers and I. Otung, "Rain fade dynamics for earth-space satellite links at V-band," presented at PREP 2005, University of Lancaster, 2005.
- [18] A. Chambers and I. Otung, "Analysis of Rain Fade Slope for Ka and V-band Satellite Links in Southern England," *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 5, pp. 1380-1387, 2004.
- [19] A. P. Chambers and I. E. Otung, "Neural network approach to short-term fade prediction on satellite links," *Electronics letters*, vol. 41, no. 23, pp. 1290-1292, 2005.

References

- [20] A. Chambers and I. Otung, "A comparison of short-term prediction methods on Ka and V-band satellite links," presented at ESA Propagation Workshop 2005 on Radiowave Propagation Modelling and Measurements for SatCom & SatNav Systems, Noordwijk, Holland, 2005.
- [21] A. Chambers and I. Otung, "Near-optimum short-term fade prediction on satellite links at Ka and V-bands," *International Journal of Satellite Communications and Networking*, vol. 26, no. 1, pp. 31-43, 2007.
- [22] B. C. Grémont, "Fade countermeasure modelling for Ka band digital satellite links," University of Coventry, 1997
- [23] L. Castanet, "Fade Mitigation Techniques for new SatCom systems operating at Ka and V bands," SUPAERO, 2001
- [24] Eutelsat Communications, <http://www.eutelsat.com>, accessed on 10th January 2007
- [25] Avanti Communications, <http://www.avanti-communications.com>, accessed on 18th November 2007
- [26] M. Filip and E. Vilar, "Implementation of adaptive modulation as a fade countermeasure," *International Journal of Satellite Communications*, vol. 12, no. -, pp. 181-191, 1994.
- [27] B. Grémont and M. Filip, "Simulation of a high frequency satellite link with a fade countermeasure," presented at IEE Conference on Antennas and Propagation, York, UK, 1999.
- [28] P. Zong and M. Filip, "Modelling of a VSAT link physical layer fade countermeasure controlled at the data link layer," *IEE Proceedings - Communications*, vol. 149, no. 05-06, pp. 277-282, 2002.
- [29] A. Malygin, M. Filip, and E. Vilar, "Neural network implementation of a fade countermeasure controller for a VSAT link," *International Journal of Satellite Communications*, vol. 20, no. 2, pp. 79-95, 2002.
- [30] I. Otung, "Optimisation of 16-QAM modulation for nonlinear satellite channels," presented at 23rd Communications Satellite Systems Conference, Rome, Italy, 2005.
- [31] A. H. Aghvami, "performance analysis of 16-ary QAM signalling through two-link nonlinear channels in additive Gaussian noise," *IEE Proceedings F (Communications, Radar and Signal Processing)*, vol. 131, no. 4, pp. 403-406, 1984.
- [32] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261-1271, 1996.
- [33] S. Benedetto and G. Montorsi, "Performance of continuous and blockwise decoded turbo codes," *IEEE Communications Letters*, vol. 1, no. 3, pp. 77-79, 1997.
- [34] M. Ferrari and S. Bellini, "Low BER turbo codes for satellite communications," presented at Seventh International Workshop on Digital Signal Processing Techniques for Space Communications, Sesimbra, Portugal, 2001.
- [35] C. Berrou, M. Jézéquel, C. Douillard, and S. Kérourédan, "Duo-binary turbo codes associated with high-order modulations," presented at ESA TTC, Noordwijk, The Netherlands, 2001.
- [36] S. Ventouras, I. Otung, and C. Wrench, "Simulation of satellite systems operating at Ka-band and above using experimental time series of tropospheric attenuation," presented at Simulation and Modelling of Satellite Systems, London, UK, 2002.

References

- [37] L. Castanet, M. Bousquet, and D. Mertens, "Simulation of the performance of a Ka-band VSAT videoconferencing system with uplink power control and data rate reduction to mitigate atmospheric propagation effects," *International Journal of Satellite Communications*, vol. 20, pp. 231-249, 2002.
- [38] C. D. Hughes and M. Tomlinson, "The use of spread-spectrum coding as a fading countermeasure at 20/30 GHz," *ESA Journal*, vol. 11, no. 4, pp. 73-81, 1988.
- [39] K. Watanabe, S. Sampei, and N. Morinaga, "An Adaptive Access Control Scheme for a reservation TDMA/Adaptive Modulation System," *Electronics and Communications in Japan, part 1*, vol. 85, no. 11, pp. 77-86, 2002.
- [40] E. Noussi, B. Grémont, and M. Filip, "Adaptive MF-TDMA: Burst length control as a rain fade countermeasure," presented at CSNDSP, 4th international symposium proceedings, Newcastle-upon-Tyne, UK, 2004.
- [41] E. Noussi, B. Grémont, and M. Filip, "A demand assignment algorithm for FMT resource management in DVB-RCS networks," presented at PREP 2005, University of Lancaster, UK, 2005.
- [42] D.-H. Kim, S.-H. Hwang, U.-Y. Pak, and K.-C. Whang, "Adaptive CDMA Scheme as a Rain Fade Countermeasure in Ka-Band Geosynchronous Satellite Communications," *IEICE Trans. Commun.*, vol. E38-B, no. 12, pp. 2600-06, 2000.
- [43] M. Luglio, R. Mancini, C. Riva, A. Paraboni, and F. Barbaliscia, "Large-scale site diversity for satellite communication networks," *International Journal of Satellite Communications*, vol. 20, pp. 251-260, 2002.
- [44] D. D. Zenobio, P. Lombardi, P. Migliorini, and E. Russo, "Site Diversity: A powerful antifading technique for satellite communications in the 20/30 GHz bands," presented at Olympus Utilisation Conference (ESA SP-292), Vienna, 1989.
- [45] E. Russo, "Implementation of a space diversity system for ka-band satellite communications," presented at ICC '93, Geneva, Switzerland, 1993.
- [46] C. Enjamio, E. Vilar, D. Ndzi, and F. Perez-Fontán, "Short-scale diversity in a dynamic rain fade environment," *International Journal of Satellite Communications and Networking*, vol. 23, pp. 143-152, 2005.
- [47] A. D. Panagopoulos, P.-D. M. Arapoglou, and P. G. Cottis, "Site versus orbital diversity: Performance comparison based on propagation characteristics at the ku band and above," *Antennas and wireless propagation letters*, vol. 3, pp. 26-29, 2004.
- [48] E. Matricciani, "Prediction of orbital diversity performance in satellite communication systems affected by rain attenuation," *International Journal of Satellite Communications*, vol. 15, pp. 45-50, 1997.
- [49] A. D. Panagopoulos and J. D. Kanellopoulos, "Prediction of triple-orbital diversity performance in Earth-space communication," *International Journal of Satellite Communications*, vol. 20, pp. 187-200, 2002.
- [50] L. Dossi, G. Tartara, and E. Matricciani, "Frequency diversity in millimeter wave satellite communications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 2, pp. 567-573, 1992.
- [51] M. Luglio, "Fade countermeasures in Ka band: application of frequency diversity to a satellite system," presented at Tenth International Conference on Digital Satellite Communications, Rome University, Italy, 1995.

References

- [52] D. R. Pauluzzi and N. C. Beaulieu, "A comparison of SNR estimation techniques for the AWGN channel," *IEEE Transactions on Communications*, vol. 48, no. 10, pp. 1681-1691, 2000.
- [53] B. C. Grémont, "Event Detection, Control and Performance Modelling," presented at 2nd international workshop of COST action 280, ESTEC, NL, 2003.
- [54] S. Egami, "Individual closed-loop satellite access power control system using overall satellite link quality level," *IEEE Transactions on Communications*, vol. 30, no. 7, pp. 1806-1808, 1982.
- [55] D. G. Sweeney and C. W. Bostian, "Implementing adaptive power control as a 30/20-GHz fade countermeasure," *IEEE Transactions on Antennas and Propagation*, vol. 47, no. 1, pp. 40-46, 1999.
- [56] S.-H. Hwang, D.-H. Kim, S.-I. Lee, and K.-C. Whang, "Power control and macrodiversity as fade countermeasures in satellite CDMA transmission at Ka-band," *IEICE Transactions on Communications*, vol. E82-B, no. 11, pp. 1878-1882, 1999.
- [57] B. Grémont, M. Filip, and E. Vilar, "Evaluation of CNR statistics for satellite link with multiple countermeasures," *Electronics letters*, vol. 36, no. 11, 2000.
- [58] Y. Karasawa and Y. Maekawa, "Ka-band Earth-space propagation research in Japan," *Proceedings of the IEEE*, vol. 85, no. 6, pp. 821-842, 1997.
- [59] P. Angeletti, M. Lisi, and R. Mizzoni, "Antennas for broadband and mobile satellite communications," presented at IEEE Antennas and Propagation Society International Symposium, Piscataway, NJ, USA, 2002.
- [60] R. J. Watson, A. Page, and P. A. Watson, "Time-series of attenuation on EHF and SHF fixed satellite links derived from meteorological and forecast data," presented at IEE Seminar Personal Broadband Satellite, London, UK, 2002.
- [61] F. S. Marzano, J. F. Turk, P. Ciotti, and S. D. Michele, "Potential of combined spaceborne infrared and microwave radiometry for near real-time rainfall attenuation monitoring along earth-satellite links," *International Journal of Satellite Communications*, vol. 19, pp. 385-412, 2001.
- [62] L. Castanet, J. Lemorton, and M. Bousquet, "Fade mitigation techniques for new SatCom services at Ku-band and above: a review.," presented at COST 255 - First International Workshop on Radiowave Propagation Modelling for SatCom Services at Ku-band and Above, Noordwijk, Netherlands, 1999.
- [63] K. Kastamonitis, B. Grémont, and M. Filip, "Online extraction of scintillations for satellite links with up-link power control," presented at 1st international workshop of COST action 280, Malvern, UK, 2002.
- [64] E. A. Newcombe and S. Pasupathy, "Error rate monitoring for digital communications," *Proceedings of the IEEE*, vol. 70, no. 8, pp. 805-828, 1982.
- [65] P. F. Hartigan and A. P. Gallois, "The use of satellite imagery for forecasting rain attenuation," presented at IEE Eighth International Conference on Antennas and Propagation, Edinburgh, 1993.
- [66] L. Dossi, "Real-time prediction of attenuation for applications to fade countermeasures in satellite communications," *Electronics letters*, vol. 26, no. 4, pp. 250-251, 1990.
- [67] K. Kastamonitis, B. Gremont, and M. Filip, "Short-term prediction of rain attenuation based on fade slope," *Electronics letters*, vol. 39, no. 8, pp. 687-689, 2003.
- [68] B. Grémont, M. Filip, P. Gallois, and S. Bate, "Comparative Analysis and Performance of Two Predictive Fade Detection Schemes for Ka-Band Fade

References

- Countermeasures," *IEEE Transactions on Selected Areas in Communications*, vol. 17, no. 2, pp. 180-192, 1999.
- [69] M. M. J. L. van de Kamp, "Short-term prediction of rain attenuation using two samples," *Electronics letters*, vol. 38, no. 23, pp. 1476-1477, 2002.
- [70] S. K. Barton and S. E. Dinwiddy, "A technique for estimating the throughput of adaptive TDMA fade countermeasure systems," *International Journal of Satellite Communications*, vol. 6, pp. 331-341, 1988.
- [71] M. Schnell and D. V. Hugo, "Fade countermeasures and adaptive resource sharing for an experimental TDMA satellite communications system operating at Ka-band," *European Transactions on Telecommunications & Related Technologies*, vol. 9, no. 6, pp. 561-568, 1998.
- [72] F. Davoli, M. Marchese, and M. Mongelli, "Resource allocation in satellite networks: certainty equivalent approaches versus sensitivity estimation algorithms," *International Journal of Communications Systems*, vol. 18, pp. 3-36, 2005.
- [73] N. Celandroni, E. Ferro, and F. Potorti, "A multi-level satellite channel allocation algorithm for real time VBR data," *International Journal of Satellite Communications*, vol. 20, pp. 47-61, 2002.
- [74] N. Celandroni, F. Davoli, E. Ferro, and A. Gotta, "Adaptive cross-layer bandwidth allocation in a rain-faded satellite environment," *International Journal of Communications Systems*, vol. 19, pp. 509-530, 2005.
- [75] N. Celandroni, E. Ferro, and F. Potorti, "DRIFS-TDMA: A proposal for a satellite access distributed-control algorithm for multimedia traffic in a faded environment," *International Journal of Satellite Communications*, vol. 15, pp. 227-235, 1997.
- [76] N. Celandroni, E. Ferro, and F. Potorti, "FEEDERS-TDMA: A distributed-control algorithm for satellite channel capacity assignment in a mixed traffic and faded environment," *International Journal of Satellite Communications*, vol. 15, pp. 185-195, 1997.
- [77] S. Ventouras, C. L. Wrench, and S. A. Callaghan, "Measurement and analysis of satellite beacon transmissions at frequencies up to 50 GHz. Part 1: Attenuation Statistics and Frequency Scaling of Attenuation Values," Project report for the Radio Communications Agency, UK September 2000, Updated March 2003.
- [78] E. Matricciani and C. Riva, "Evaluation of the feasibility of satellite systems design in the 10-100 GHz frequency range," *International Journal of Satellite Communications*, vol. 16, pp. 237-247, 1998.
- [79] S. Ventouras, C. L. Wrench, and S. A. Callaghan, "Measurement and analysis of satellite beacon transmissions at frequencies up to 50 GHz - Part 2: Fade and inter-fade duration," Project report for the Radio Communications Agency, UK 2001.
- [80] F. Dintelman, "Analysis of 11 GHz slant path fade duration and fade slope," *Electronics letters*, vol. 17, no. 7, pp. 267-268, 1981.
- [81] E. Matricciani, "Rate of change of signal attenuation from Sirio at 11.6 GHz," *Electronics letters*, vol. 17, no. 3, pp. 139-141, 1981.
- [82] E. Matricciani, "Effects of filtering on rate of change of rain-induced attenuation," *Electronics letters*, vol. 18, no. 11, pp. 477-478, 1982.
- [83] W. L. Stutzman, T. Pratt, A. Safaai-Jazi, P. W. Remaklus, J. Laster, B. Nelson, and H. Ajaz, "Results from the Virginia Tech Propagation Experiment Using the Olympus Satellite 12, 20 and 30 GHz Beacons," *IEEE Transactions on Antennas and Propagation*, vol. 43, no. 1, pp. 54-62, 1995.

References

- [84] B. Nelson and W. L. Stutzman, "Fade slope on 10 to 30 GHz earth-space communication links - measurements and modelling," *IEE Proceedings on Microwaves, Antennas and Propagation*, vol. 143, no. 04, pp. 353-357, 1996.
- [85] L. v. Coevering, K. G. Holleboom, G. Brussaard, and J. Dijk, "Some results of the Olympus propagation experiment at Eindhoven University of Technology," presented at Ninth International Conference on Antennas and Propagation, Eindhoven, Netherlands, 1995.
- [86] S. Ventouras, P. G. Davies, and J. R. Norbury, "Olympus-based propagation measurements in southern UK," presented at IEE ICAP '95, 1995.
- [87] J. Feil, L. J. Ippolito, H. Helmken, C. E. Mayer, S. Horan, and R. E. Henning, "Fade slope analysis for Alaska, Florida and New Mexico ACTS propagation data at 20 and 27.5 GHz," *Proceedings of the IEEE*, vol. 85, no. 6, pp. 926-935, 1997.
- [88] M. M. J. van de Kamp, "Statistical Analysis of Rain Fade Slope," *IEEE Transactions on Antennas and Propagation*, vol. 51, no. 8, pp. 1750-1759, 2003.
- [89] "Prediction method of fade dynamics on Earth-space paths," in *Recommendations of the ITU-R*, vol. P.1623-1, 2005.
- [90] S. Ventouras, S. A. Callaghan, and C. L. Wrench, "Results and implications from ITALSAT propagation experiment measurements at the CCLRC-Chilbolton observatory," presented at ICAP 2003, Exeter, UK, 2003.
- [91] T. Palade and C. Riva, "ITALSAT propagation experiment at 18.7, 39.6 and 49.5 GHz at Spino d'Adda: Seven years of fade slope statistics," presented at 1st international workshop of COST action 280, Malvern, UK, 2002.
- [92] M. Schnell and U.-C. Fiebig, "Fade slope statistics of 40 GHz beacon signals," *Electronics letters*, vol. 33, no. 21, 1997.
- [93] K. Kastamonitis, B. Gremont, and M. Fillip, "Empirical study of rain fade slope conditional statistics on satellite links," presented at Twelfth International Conference on Antennas and Propagation (ICAP 2003), Exeter, UK, 2003.
- [94] P. D. Baxter, G. J. G. Upton, and D. Eden, "Revised method for calculation of rain-fade-slope," *Electronics letters*, vol. 37, no. 10, 2001.
- [95] I. E. Otung, M. O. Al-Nuaimi, and B. G. Evans, "Extracting Scintillations from satellite Beacon Propagation Data," *IEEE Transactions on Antennas and Propagation*, vol. 46, no. 10, pp. 1580-1581, 1998.
- [96] P. D. Baxter, G. J. G. Upton, and D. Eden, "Measuring rain-fade-slope: Objective evaluation of filtering techniques," presented at 2nd international workshop of COST action 280, ESTEC, NL, 2003.
- [97] "Prediction method of fade dynamics on Earth-space paths," in *Recommendations of the ITU-R*, vol. P.1623, 2003.
- [98] D. Parker M and H. Johnson R, "Structures and dynamics of Quasi-2D mesoscale convective systems," *Journal of Atmospheric Sciences*, vol. 61, no. 5, pp. 545-567, 2004.
- [99] K. I. Timothy, J. T. Ong, and E. B. L. Choo, "Descriptive fade slope statistics on INTEL SAT Ku-band communication link," *Electronics letters*, vol. 36, no. 16, 2000.
- [100] P. A. Watson, M. J. Leita, V. Sathiaselan, M. Gunes, J. P. V. Polares Baptista, B. A. Potter, N. Sengupta, O. Turney, and G. Brussaard, "Prediction of attenuation on satellite-earth links in the European region," *IEE Proceedings*, vol. 134, no. 6, pp. 583-596, 1987.

References

- [101] A. Paraboni and C. Riva, "A new method for the prediction of fade duration statistics in satellite links above 10 GHz," *International Journal of Satellite Communications*, vol. 12, pp. 387-394, 1994.
- [102] J. D. Laster and W. L. Stutzman, "Frequency scaling of rain attenuation for satellite communication links," *IEEE Transactions on Antennas and Propagation*, vol. 43, no. 11, pp. 1207-1216, 1995.
- [103] B. C. Grémont and M. Filip, "Modelling and applications of the Instantaneous Frequency Scaling Factor (IFSF) of Rain Attenuation," presented at COST 255 First International Workshop on Radiowave Propagation Modelling for SatCom Services at Ku-band and above, Noordwijk, Netherlands, 1998.
- [104] M. M. J. L. van de Kamp and L. Castanet, "Fade Dynamics Review," presented at Cost Action 280 First International Workshop, Malvern, 2002.
- [105] C. Berrou and M. Jézéquel, "Non-binary convolutional codes for turbo coding," *Electronics letters*, vol. 35, no. 1, 1999.
- [106] S. K. Shin, S. I. Lee, and S. P. Lee, "Evaluation of block turbo code performance with the reduced search trellis decoding method," *IEE Proceedings*, vol. 148, no. 3, pp. 125-131, 2001.
- [107] A. Banerjee, F. Vatta, B. Scanavino, and J. Daniel J. Costello, "Nonsystematic turbo codes," *IEEE Transactions on Communications*, vol. 53, no. 11, pp. 1841-1849, 2005.
- [108] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. IT-20, pp. 248-287, 1974.
- [109] M. C. Valenti and J. Sun, "The UMTS turbo code and an efficient decoder implementation suitable for software defined radios," *International Journal of Wireless Information Networks*, vol. 8, pp. 203-216, 2001.
- [110] M. C. Valenti, S. Cheng, and R. I. Seshadri, "Turbo and LDPC codes for digital video broadcasting," in *Turbo Code Applications: A Journey from a Paper to Realization*: Springer, 2005.
- [111] D. Giancrisofaro and Bartolazzi, "Novel DVB-RCS standard Turbo code: details and performances of a decoding algorithm," presented at Seventh International Workshop on Digital Signal Processing Techniques for Space Communications, Sesimbra, Portugal, 2001.
- [112] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Transactions on Telecommunications & Related Technologies*, vol. 8, no. 2, pp. 119-125, 1997.
- [113] A. J. Viterbi, "An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 260-264, 1998.
- [114] J. B. Anderson and S. M. Hladik, "Tailbiting MAP Decoders," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 297-302, 1998.
- [115] Iterative Solutions, <http://www.iterativesolutions.com>, accessed on 18th January 2006
- [116] C. Douillard, M. Jézéquel, C. Berrou, N. Brengarth, J. Tusch, and N. Pham, "The turbo code standard for DVB-RCS," presented at 2nd Int'l Symposium on Turbo Codes & Related Topics, 2000.
- [117] M. M. J. L. van de Kamp, "Rain Attenuation as a Markov Process: How to make an event," presented at 2nd International Workshop of COST Action 280, ESTEC, NL, 2003.

References

- [118] B. C. Grémont and M. Filip, "Spatio-temporal rain attenuation model for application to fade mitigation techniques," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 5, pp. 1245-1256, 2004.
- [119] R. K. Crane, *Electromagnetic wave propagation through rain*: Wiley, 1996.
- [120] E. Matricciani and A. P. Bonati, "Rain cell size statistics inferred from long term point rain rate: Model and Results," presented at Ka-band utilization conference, Sorrento, Italy, 1997.
- [121] B. Grémont, A. P. Gallois, and S. D. Bate, "Efficient fade compensation for Ka band VSAT systems," presented at 2nd Ka band Utilisation Conference and International Workshop SCGII, Florence, Italy, 1996.
- [122] L. Ljung, *System Identification - Theory for the User*, 2nd ed: Prentice Hall, 1999.
- [123] K. J. Åström, *Introduction to Stochastic Control Theory*, vol. 70. New York and London: Academic Press, 1970.
- [124] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, 1st ed. New York: Prentice-Hall, 1985.
- [125] S. Ventouras, C. L. Wrench, and S. A. Callaghan, "New thinking required to offset limitations imposed by V-band propagation," presented at 19th AIAA Int. Comm. Sat. Sys. Conf., Toulouse, France, 2001.
- [126] I. E. Otung, "Scintillation fade and enhancement duration on a 20GHz satellite downlink," *Electronics letters*, vol. 32, no. 14, pp. 1319-1321, 1996.
- [127] I. E. Otung, "Short term distribution of amplitude scintillation on a satellite link," *Electronics letters*, vol. 31, no. 16, pp. 1328-1329, 1995.
- [128] J. M. Woodroffe, P. G. Davies, and J. R. Norbury, "Combined Olympus and ITALSAT propagation experiment in the UK," presented at WPP-60 Proc. Olympus utilisation conference, Seville, Spain, 1993.
- [129] "Reference Standard Atmosphere for Gaseous Attenuation," in *Recommendations of the ITU-R*, vol. PN.835-1, 1994.
- [130] L. H.J, "An Atmospheric Millimeter-wave Propagation Model," *International Journal of Infrared and Millimeter Waves*, vol. 10, no. 6, pp. 631-6, 1989.
- [131] I. E. Otung and A. Savvaris, "Estimating tropospheric scintillation intensity on earth-space propagation paths," *Electronics letters*, vol. 42, no. 7, pp. 381-382, 2006.
- [132] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1-15, 1994.
- [133] Nielsen Netratings, <http://www.nielsen-netratings.com>, accessed on 10th January 2007
- [134] P. Barsocchi, N. Celandroni, F. Davoli, E. Ferro, G. Giambene, F. J. G. Castaño, A. Gotta, J. I. Moreno, and P. Todorova, "Radio resource management across multiple protocol layers in satellite networks: A tutorial overview," *International Journal of Satellite Communications and Networking*, vol. 23, pp. 265-305, 2005.
- [135] ATLANTIC BIRD 1 Satellite data sheet, accessed on December 2005 2005
- [136] B. Sklar, "Channel coding: part 2," in *Digital Communications*: Prentice Hall, 2001.
- [137] M. C. Valenti, S. Cheng, and R. I. Seshadri, *Turbo Code Applications: A Journey from a Paper to Realization*: Springer, 2005.

References

- [138] Y. Suzuki, M. Shimada, H. Mineno, T. Shinozuka, H. Kuroiwa, and H. Inomata, "TDMA site diversity switching experiments with Japanese CS," *IEEE Journal on Selected Areas in Communications*, vol. SAC-1, no. 4, 1983.
- [139] N. Celandroni, "Comparison between distributed and centralized demand assignment TDMA satellite access schemes," *International Journal of Satellite Communications*, vol. 14, pp. 95-111, 1996.
- [140] A. Iera, A. Molinaro, P. Pace, and S. Marano, "Dimensioning and Effective Handling of Signalling Channels in a Multimedia GEO Satellite Platform," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 2, pp. 550-566, 2005.
- [141] A. Iuoras, P. Takats, C. Black, R. DiGirolamo, E. A. Wibowo, J. Lambadaris, and M. Devetsikiotis, "Quality of service-oriented protocols for resource management in packet switched satellites," *International Journal of Satellite Communications*, vol. 17, pp. 129-141, 1999.
- [142] N. Celandroni, F. Davoli, E. Ferro, and A. Gotta, "A combined resource allocation method for fade countermeasure and call admission control in satellite networks," presented at 2nd international workshop of COST action 280, ESTEC, Netherlands, 2003.

Appendices

Appendix A

ITU-R model for rain fade slope

The ITU-R model [97] calculates the standard deviation of the conditional fade slope at a given attenuation as:

$$\sigma_{\zeta} = sF(f_B, \Delta t)A \quad (\text{dB/s}) \quad (\text{A1})$$

where A is the attenuation level, s is assumed constant for geographical location, and averages at 0.01 in Europe and the United States and $F(f_B, \Delta t)$ gives the dependence on time interval Δt and the cut-off frequency of the low pass filter f_B where:

$$F(f_B, \Delta t) = \frac{\sqrt{2\pi^2}}{\sqrt{\left(\frac{1}{f_B^b} + (2\Delta t)^b\right)^{1/b}}} \quad (\text{dB/s}) \quad (\text{A2})$$

where $b = 2.3$.

The probability density p of fade slope ζ at a given attenuation value is:

$$p(\zeta | A) = \frac{2}{\pi\sigma_{\zeta} \left(1 + \left(\frac{\zeta}{\sigma_{\zeta}}\right)^2\right)^2} \quad (\text{dB/s}) \quad (\text{A3})$$

The probability that at a given attenuation value, a fade slope ζ is exceeded is:

$$P(\zeta | A) = \frac{1}{2} - \frac{\zeta/\sigma_{\zeta}}{\pi(1 + (\zeta/\sigma_{\zeta})^2)} - \frac{\arctan(\zeta/\sigma_{\zeta})}{\pi} \quad (\text{A4})$$

Appendix B

Turbo Decoding Process M-file

```
function y = turbodecodeDVB(x,Ni,rate,it,qbits,data)
% y = turbodecodeDVB(x,Ni,rate,it,qbits,data) decodes the 4-state LLRs in x using a
% SISO APP module where x are the bit-wise LLRs, Ni is the block size, rate is the
coding rate, it is the number of iterations desired, qbits is currently a placeholder
for number of quantization bits and data is the original data sequence for comparison
and early termination if the errors are corrected for decoding speed improvement.

[PO,P1,P2,P3,N] = GenPermParams(Ni);
k = 2*N;
ylen = length(x);
M = getM(N,rate);
blocksize = 2*(N + M);
Nblocks = length(x)/blocksize;
acc = 0.001;
errors = zeros(it,1);
errors_pre = zeros(it,1);
y = zeros(Nblocks*k,1);

for c1 = 1:Nblocks

    % Depuncture code
    dataword = x(c1*blocksize-blocksize+1:c1*blocksize-2*M);
    xtrabits = x(c1*blocksize-blocksize+k+1:c1*blocksize);
    Y = zeros(2,N);
    W = zeros(2,N);

    switch rate
        case '1/3'
            for bcnt = 1:1:N
                Y(:,bcnt) = xtrabits(bcnt*2-1:bcnt*2);
                W(:,bcnt) = xtrabits(bcnt*2-1+2*N:bcnt*2+2*N);
            end
        case '2/5'
            for bcnt = 1:2:N
                wcnt = fix(bcnt/2+1);
                Y(:,bcnt) = xtrabits(bcnt*2-1:bcnt*2);
                Y(:,bcnt+1) = xtrabits(bcnt*2+1:bcnt*2+2);
                W(:,bcnt) = xtrabits(wcnt*2-1+2*N:wcnt*2+2*N);
            end
        case '1/2'
            for bcnt = 1:1:N
                Y(:,bcnt) = xtrabits(bcnt*2-1:bcnt*2);
            end
        case '2/3'
            for bcnt = 1:2:N
                wcnt = fix(bcnt/2+1);
                Y(:,bcnt) = xtrabits(wcnt*2-1:wcnt*2);
            end
        case '3/4'
            for bcnt = 1:3:N
                wcnt = fix(bcnt/3+1);
                Y(:,bcnt) = xtrabits(wcnt*2-1:wcnt*2);
            end
        case '4/5'
            for bcnt = 1:4:N
                wcnt = fix(bcnt/4+1);
                Y(:,bcnt) = xtrabits(wcnt*2-1:wcnt*2);
            end
        case '6/7'
            for bcnt = 1:6:N
                wcnt = fix(bcnt/6+1);
                Y(:,bcnt) = xtrabits(wcnt*2-1:wcnt*2);
            end
        otherwise
            disp('Error: Invalid code rate');
            disp('Code rates can be 1/3, 2/5, 1/2, 2/3, 3/4, 4/5 & 6/7');
            return;
    end
end
```

```

end

LLRu = zeros(3,N);
ext = zeros(3,N);
LLRcnt(1:4:N*4) = dataword(1:2:N*2);
LLRcnt(2:4:N*4) = dataword(2:2:N*2);
LLRcnt(3:4:N*4) = W(1,:);
LLRcnt(4:4:N*4) = Y(1,:);
datawordperm = interleaveDVB(dataword(1:N*2),N,P0,P1,P2,P3);
LLRcit(1:4:N*4) = datawordperm(1:2:N*2);
LLRcit(2:4:N*4) = datawordperm(2:2:N*2);
LLRcit(3:4:N*4) = W(2,:);
LLRcit(4:4:N*4) = Y(2,:);
LLRcn = bit2llrmex(LLRcnt,2,'DVB');
LLRci = bit2llrmex(LLRcit,2,'DVB');

% do first stage
LLR1 = sisoDecodeDVBmex(ext,LLRcn);
ext = interleaveLLRDVBmex(LLR1-ext,N,P0,P1,P2,P3);
LLR2 = deinterleaveLLRDVBmex(sisoDecodeDVBmex(ext,LLRci),N,P0,P1,P2,P3);
output = Somap([zeros(1,N);LLR2],1)';
errtmp = sum(xor(fix((sign(output)+1)/2),data(c1*k-k+1:c1*k)));
ext = LLR2-deinterleaveLLRDVBmex(ext,N,P0,P1,P2,P3);
% do stages 2 to 8 unless errors are all corrected
c=2;
while c <= it && errtmp
    LLR1 = sisoDecodeDVBmex(ext,LLRcn);
    ext = interleaveLLRDVBmex(LLR1-ext,N,P0,P1,P2,P3);
    LLR2 = deinterleaveLLRDVBmex(sisoDecodeDVBmex(ext,LLRci),N,P0,P1,P2,P3);
    output = Somap([zeros(1,N);LLR2],1)';
    errtmp = sum(xor(fix((sign(output)+1)/2),data(c1*k-k+1:c1*k)));
    ext = LLR2-deinterleaveLLRDVBmex(ext,N,P0,P1,P2,P3);
    c=c+1;
end

y(c1*k-k+1:c1*k) = output;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function M = getM(N,rate)
% Returns the number of code couples

switch rate
case '1/3'
    M = 2*N;
case '2/5'
    M = 3*N/2;
case '1/2'
    M = N;
case '2/3'
    M = N/2;
case '3/4'
    switch mod(N,3)
    case 0; M = N/3;
    case 1; M = (N-4)/3 + 2;
    case 2; M = (N-8)/3 + 3;
    end
case '4/5'
    M = N/4;
case '6/7'
    switch mod(N,3)
    case 0; M = N/6;
    case 1; M = (N-4)/6 + 1;
    case 2; M = (N-8)/6 + 2;
    end
otherwise
    disp('Error: Invalid code rate');
    disp('Code rates can be 1/3, 2/5, 1/2, 2/3, 3/4, 4/5 & 6/7');
    return;
end

```

Appendix C1

The Decoder Block Mex File

```
#include <math.h>
#include <mex.h>
#include <matrix.h>
#include <stdlib.h>
#include "sisoDVB.h"

void mexFunction(
    int             nlhs,
    mxArray         *plhs[],
    int             nrhs,
    const mxArray   *prhs[] )
{
    int N = mxGetN(prhs[0]);
    sisoDVB(mxGetPr(plhs[0]=mxCreateDoubleMatrix(3,N,mxREAL)),mxGetPr(prhs[0]),mxGetPr(prhs[1]),N);
    return;
}
```

Appendix C2

The Decoder Block Header File

```
#define maxLLR 1e7

static double maxa(double P[], int l)
{
    int c;
    double x = -maxLLR;
    for(c=0;c<l;c++)
    {
        if(P[c]>x)
            x = P[c];
    }
    return x;
}

static void sisoDVB(double LLRuo[], double LLRui[], double LLRci[],
                    int lenu)
{
    int inSymbols = 4;
    int outSymbols = 16;
    int Nstates = 8;
    int nextStates[] = {0,4,1,5,6,2,7,3,
                        7,3,6,2,1,5,0,4,
                        4,0,5,1,2,6,3,7,
                        3,7,2,6,5,1,4,0};
    int outputs[] = {0, 0, 1, 1, 3, 3, 2, 2,
                    7, 7, 6, 6, 4, 4, 5, 5,
                    11,11,10,10,8, 8, 9, 9,
                    12,12,13,13,15,15,14,14};

    int inBits = 2;
    int outBits = 4;
    int codeSymbols = 1 << (outBits-inBits);
    int ratio = 2;
    int lenc = ratio*lenu;
    int na,nb,st,insym,cin,cout,stopa,stopb;
    double dum,da,db;
    double cstop = 0.1;
    double P[4];
    double tmpP[2];
    double Atmp[8];
    double Btmp[8];
    double PN[8];
    double *Gamma, *Alpha, *Beta;

    // Initialise internal arrays
    Gamma = mxCalloc(outSymbols*lenu, sizeof(double));
    Alpha = mxCalloc(Nstates*(lenu+1), sizeof(double));
    Beta = mxCalloc(Nstates*(lenu+1), sizeof(double));
    for(na=0;na<8;na++)
        Alpha[na] = -2.0;
    for(na=8;na<Nstates*(lenu+1);na++)
        Alpha[na] = -maxLLR;
    for(nb=Nstates*lenu;nb<Nstates*(lenu+1);nb++)
        Beta[nb] = -2.0;

    // Calculate all possible branch metrics (Gamma)
    for(nb=lenu-1;nb>=0;nb--)
    {
        for(cin=0;cin<inSymbols;cin++)
        {
            for(cout=0;cout<codeSymbols;cout++)
            {
                Gamma[nb*outSymbols+cin*codeSymbols+cout] = 0.0;
                if(cin > 0)
                    Gamma[nb*outSymbols+cin*codeSymbols+cout] += LLRui[nb*3+cin-
1]+LLRci[nb*6+cin-1];
                if(cout > 0)
                    Gamma[nb*outSymbols+cin*codeSymbols+cout] += LLRci[nb*6+cout+2];
            }
        }
    }
}
```

```

    }
}

// Calculate Alpha & Beta state metrics
for(na=1;na<lenu+1;na++)
{
    nb = lenu-na;
    for(st=0;st<Nstates;st++)
    {
        for(insym=0;insym<inSymbols;insym++)
        {
            P[insym] = Beta[(nb+1)*Nstates+nextStates[Nstates*insym+st]] +
Gamma[nb*outSymbols+outputs[Nstates*insym+st]];
            tmpP[0] = Alpha[(na-1)*Nstates+st] + Gamma[(na-
1)*outSymbols+outputs[Nstates*insym+st]];
            tmpP[1] = Alpha[na*Nstates+nextStates[Nstates*insym+st]];
            Alpha[na*Nstates+nextStates[Nstates*insym+st]] = maxa(tmpP,2);
        }
        Beta[nb*Nstates+st] = maxa(P,inSymbols);
    }
    // Normalise Alphas
    for(st=1;st<Nstates;st++)
        Alpha[na*Nstates+st] = Alpha[na*Nstates+st] - Alpha[na*Nstates];
    Alpha[na*Nstates] = 0.0;
    // Normalise Betas
    for(st=1;st<Nstates;st++)
        Beta[nb*Nstates+st] = Beta[nb*Nstates+st] - Beta[nb*Nstates];
    Beta[nb*Nstates] = 0.0;
}

// Wrap around trellis until approximate match for Alphas and Betas (tailbiting)
na = 1;
nb = lenu-1;
stopa = 1;
stopb = 1;
for(st=0;st<Nstates;st++)
{
    Btmp[st] = -maxLLR;
    Alpha[st] = Alpha[lenu*Nstates+st];
    Beta[lenu*Nstates+st] = Beta[st];
}
while(stopa||stopb)
{
    for(st=0;st<Nstates;st++)
        Atmp[st] = -maxLLR;
    for(st=0;st<Nstates;st++)
    {
        for(insym=0;insym<inSymbols;insym++)
        {
            P[insym] = Beta[(nb+1)*Nstates+nextStates[Nstates*insym+st]] +
Gamma[nb*outSymbols+outputs[Nstates*insym+st]];
            tmpP[0] = Alpha[(na-1)*Nstates+st] + Gamma[(na-
1)*outSymbols+outputs[Nstates*insym+st]];
            tmpP[1] = Atmp[nextStates[Nstates*insym+st]];
            Atmp[nextStates[Nstates*insym+st]] = maxa(tmpP,2);
        }
        Btmp[st] = maxa(P,inSymbols);
    }
    // Normalise Alphas
    for(st=1;st<Nstates;st++)
        Atmp[st] = Atmp[st] - Atmp[0];
    Atmp[0] = 0;
    // Normalise Betas
    for(st=1;st<Nstates;st++)
        Btmp[st] = Btmp[st] - Btmp[0];
    Btmp[0] = 0;
    // Replace or set stop flag
    da = 0.0;
    db = 0.0;
    for(st=0;st<Nstates;st++)
    {
        dum = fabs(Alpha[na*Nstates+st]-Atmp[st]);
        if(dum > da)
            da = dum;
        dum = fabs(Beta[nb*Nstates+st]-Btmp[st]);
        if(dum > db)
            db = dum;
    }
}

```

```

    }
    if(da > cstop)
    {
        for(st=0;st<Nstates;st++)
            Alpha[na*Nstates+st] = Atmp[st];
    }
    else
    {
        stopa = 0;
    }
    if(db > cstop)
    {
        for(st=0;st<Nstates;st++)
            Beta[nb*Nstates+st] = Btmp[st];
    }
    else
    {
        stopb = 0;
    }
    // Wrap again if necessary
    if(na==lenu)
    {
        na = 0;
        nb = lenu;
        cstop = cstop + 0.1;
        for(st=0;st<Nstates;st++)
        {
            Alpha[st] = Alpha[nb*Nstates+st];
            Beta[nb*Nstates+st] = Beta[st];
        }
        //mexPrintf("Wrapped again, stopa=%d, stopb=%d\n",stopa,stopb);
    }
    // Increment counters
    na++;
    nb--;
    //wrapdepth++;
    //if(wrapdepth++ > lenu*10)
    //    mexErrMsgTxt("Possible infinite loop");
}

// Calculate LLRs
for(na=0;na<lenu;na++)
{
    for(insym=0;insym<inSymbols;insym++)
    {
        for(st=0;st<Nstates;st++)
        {
            PN[st] = Alpha[na*Nstates+st] +
Beta[(na+1)*Nstates+nextStates[insym*Nstates+st]] +
Gamma[na*outSymbols+outputs[Nstates*insym+st]];
        }
        P[insym] = maxa(PN,8);
    }
    LLRuo[na*3] = P[1]-P[0];
    LLRuo[na*3+1] = P[2]-P[0];
    LLRuo[na*3+2] = P[3]-P[0];
}

mxFree(Gamma);
mxFree(Alpha);
mxFree(Beta);
}

```

Appendix D1

Interleaver Block Mex File

```
#include <math.h>
#include <mex.h>
#include <matrix.h>
#include <stdlib.h>
#include "interleaveLLRDVB.h"

void mexFunction(
    int             nlhs,
    mxArray         *plhs[],
    int             nrhs,
    const mxArray  *prhs[] )
{
    int N = mxGetN(prhs[0]);

    interleaveLLRDVB(mxGetPr(plhs[0]=mxCreateDoubleMatrix(3,N,mxREAL)),mxGetPr(prhs[0]),*m
xGetPr(prhs[1]),*mxGetPr(prhs[2]),*mxGetPr(prhs[3]),*mxGetPr(prhs[4]),*mxGetPr(prhs[5]
));
    return;
}
```


Appendix D2

Interleaver Block Header File

```
static void interleaveLLRDVB(double y[], double x[], int N, int P0,  
                             int P1, int P2, int P3)  
{  
    int i,j,P;  
    double tmp;  
  
    for(j=0;j<N;j++)  
    {  
        if(j%2==0)  
        {  
            tmp = x[j*3];  
            x[j*3] = x[j*3+1];  
            x[j*3+1] = tmp;  
        }  
    }  
  
    for(j=0;j<N;j++)  
    {  
        switch(j%4)  
        {  
            case 0:  
                P = 0;  
                break;  
            case 1:  
                P = N/2 + P1;  
                break;  
            case 2:  
                P = P2;  
                break;  
            case 3:  
                P = N/2 + P3;  
                break;  
        }  
        i = (P0*j + P + 1)%N;  
        y[i*3] = x[j*3];  
        y[i*3+1] = x[j*3+1];  
        y[i*3+2] = x[j*3+2];  
    }  
}
```

Appendix E1

Deinterleaver Block Mex File

```
#include <math.h>
#include <mex.h>
#include <matrix.h>
#include <stdlib.h>
#include "deinterleaveLLRDVB.h"

void mexFunction(
    int          nlhs,
    mxArray      *plhs[],
    int          nrhs,
    const mxArray *prhs[] )
{
    int N = mxGetN(prhs[0]);

    deinterleaveLLRDVB(mxGetPr(plhs[0]=mxCreateDoubleMatrix(3,N,mxREAL)),mxGetPr(prhs[0]),
        *mxGetPr(prhs[1]), *mxGetPr(prhs[2]), *mxGetPr(prhs[3]), *mxGetPr(prhs[4]), *mxGetPr(prhs[
5]));
    return;
}
```

Appendix E2

Deinterleaver Block Header File

```
static void deinterleaveLLRDVB(double y[], double x[], int N, int P0,  
                               int P1, int P2, int P3)  
{  
    int i,j,P;  
    double tmp;  
  
    for(j=0;j<N;j++)  
    {  
        switch(j%4)  
        {  
            case 0:  
                P = 0;  
                break;  
            case 1:  
                P = N/2 + P1;  
                break;  
            case 2:  
                P = P2;  
                break;  
            case 3:  
                P = N/2 + P3;  
                break;  
        }  
        i = (P0*j + P + 1)%N;  
        y[j*3] = x[i*3];  
        y[j*3+1] = x[i*3+1];  
        y[j*3+2] = x[i*3+2];  
    }  
  
    for(j=0;j<N;j++)  
    {  
        if(j%2==0)  
        {  
            tmp = y[j*3];  
            y[j*3] = y[j*3+1];  
            y[j*3+1] = tmp;  
        }  
    }  
}
```

Appendix F

Permutation Look-up M-File

```
function [P0,P1,P2,P3,No] = GenPermParams(Ni)
% [P0,P1,P2,P3] = GenPermParams(N) generates the P permutation parameters
% for a turbo code as described in ETSI EN 301 790 V1.3.1 - Digital Video
% Broadcasting (DVB); Interaction channel for satellite distribution
% systems. N is a multiple of 4, but the function will round any number
% down to the closest couple.

if Ni < 64
    No = 48; P0 = 11; P1 = 24; P2 = 0; P3 = 24;
elseif Ni < 212
    No = 64; P0 = 7; P1 = 34; P2 = 32; P3 = 2;
elseif Ni < 220
    No = 212; P0 = 13; P1 = 106; P2 = 108; P3 = 2;
elseif Ni < 228
    No = 220; P0 = 23; P1 = 112; P2 = 4; P3 = 116;
elseif Ni < 424
    No = 228; P0 = 17; P1 = 116; P2 = 72; P3 = 188;
elseif Ni < 432
    No = 424; P0 = 11; P1 = 6; P2 = 8; P3 = 2;
elseif Ni < 440
    No = 432; P0 = 13; P1 = 0; P2 = 4; P3 = 8;
elseif Ni < 752
    No = 440; P0 = 13; P1 = 10; P2 = 4; P3 = 2;
elseif Ni < 848
    No = 752; P0 = 19; P1 = 376; P2 = 224; P3 = 600;
elseif Ni < 856
    No = 848; P0 = 19; P1 = 2; P2 = 16; P3 = 6;
elseif Ni < 864
    No = 856; P0 = 19; P1 = 428; P2 = 224; P3 = 652;
else
    No = 864; P0 = 19; P1 = 2; P2 = 16; P3 = 6;
end

if No ~= Ni
    disp('Warning: Invalid block size, rounding to nearest');
end
```

Appendix G1

Bit-wise LLRs to Couple-wise LLRs Mex File

```
#include <math.h>
#include <mex.h>
#include <matrix.h>
#include <stdlib.h>
#include "bit2llr.h"
/* plhs[0] is function output of symbol LLRs
 * prhs[0] is bit LLRs
 * prhs[1] is the number of bits per symbol
 * prhs[2] is the flag text, 'DVB' outputs symbol-1 by N normalised
 *          values, anything other string outputs symbol by N values.
 */

void mexFunction(
    int          nlhs,
    mxArray      *plhs[],
    int          nrhs,
    const mxArray *prhs[] )
{
    int inBits = (int)*mxGetPr(prhs[1]);
    int N = mxGetN(prhs[0])/inBits; // Calculate N
    mxChar *flagtxt = mxGetChars(prhs[2]);
    int flag = (((flagtxt[0]=='D')&&(flagtxt[1]=='V')&&(flagtxt[2]=='B')) ? 1 : 0);
    //int flag = (int)*mxGetPr(prhs[2]);
    int outWidth = ((flag) ? 3 : 4);

    bit2llr(mxGetPr(plhs[0]=mxCreateDoubleMatrix(outWidth,N,mxREAL)),mxGetPr(prhs[0]),inBits,N,flag);
    return;
}
```

Appendix G2

Bit-wise LLRs to Couple-wise LLRs Header File

```
/*
 * If flag is 0 the full LLR is output, if it is 1, the normalised version
 * (inSymbols-1 rows) is sent, normalisation being to the P(0,0) symbol
 */

static void bit2llr(double y[], double x[], int inBits, int N, int flag)
{
    int inSymbols = 1 << inBits; // 2^nbits
    int outSymbols = ((flag) ? inSymbols-1 : inSymbols);
    int mod34 = inSymbols-outSymbols;
    int bitArray, insym, nb, c;

    for (c=0; c<N; c++)
    {
        for (insym=1; insym<inSymbols; insym++)
        {
            bitArray = 1 << (inBits-1);
            for (nb=0; nb<inBits; nb++)
            {
                if (bitArray & insym)
                {
                    y[c*outSymbols+insym-mod34] = y[c*outSymbols+insym-mod34] +
x[c*inBits+nb];
                }
                bitArray = bitArray >> 1;
            }
        }
    }
}
```

Appendix H

Noisy Symbol to Couple-wise LLRs M-File

```
function y = mod2llr(x,const,EsNo,Nstates,flag)

len = length(x);
y = zeros(Nstates,len);
if nargin < 5
    flag = 'Normal';
end

for ii = 1:len
    for jj = 1:Nstates
        y(jj,ii) = -EsNo*(abs(x(ii)-const(jj))^2);
    end
end

if strcmp(flag,'DVB')
    den = y(1,:);
    y(1,:) = y(2,:) - den;
    y(2,:) = y(3,:) - den;
    y(3,:) = y(4,:) - den;
    y(4,:) = [];
end
```

Appendix I

Best-fit Slope Predictor

```
function [p,e] = dossi(y,num,k,ds,fs)
% Returns the predicted value of the next sample by the dossi method of 1st
% order linear regression with offset

if nargin < 3
    fs = 1;
    ds = 1;
    k = 1;
elseif nargin < 4
    fs = 1;
    ds = 1;
elseif nargin < 5
    fs = 1;
end

n = 1;
len = length(y);
if fs/ds ~= 1;
    for c = 1:fs/ds:len
        z(n) = y(c);
        n = n + 1;
    end
else
    z = y;
end

len = length(z);

p = zeros(len,1);
if num > 1
    for c = 2:num
        pol = polyfit((1:c)',z(1:c),1);
        p(c+k) = z(c) + k*pol(1);
    end
end
for c = (num+1):(len-k)
    pol = polyfit((c-num:c)',z((c-num):c),1);
    p(c+k) = z(c) + k*pol(1);
end
e = (z-p);
```


Appendix J

Mean Slope Predictor

```
function [p,e] = kasta(y,num,k,ds,fs)
% Returns the predicted value of the next sample by the kastamonitis method

if nargin < 3
    fs = 1;
    ds = 1;
    k = 1;
elseif nargin < 4
    fs = 1;
    ds = 1;
elseif nargin < 5
    fs = 1;
end

n = 1;
len = length(y);
if fs/ds ~= 1;
    for c = 1:fs/ds:len
        z(n) = y(c);
        n = n + 1;
    end
else
    z = y;
end

len = length(z);
p = zeros(len,1);
if num > 1
    for c = 2:num
        slope = mean(z(2:c)-z(1:c-1));
        p(c+k) = z(c) + k*slope;
    end
end
for c = (num+1):(len-k)
    slope = mean(z(c-num+1:c)-z(c-num:c-1));
    p(c+k) = z(c) + k*slope;
end
e = z - p;
```

Appendix K

ARMA Predictor

```
#include "mex.h"

#define MPRES 2.2204e-16

void polystabc(double *thetah, int n)
{
    // Declarations
    mxArray *in, *out;
    double *c, *d;
    int i;

    // Create matrix and pointer to it
    in = mxCreateDoubleMatrix(n+1,1,mxREAL);
    c = mxGetPr(in);

    // Fill matrix
    c[0] = 1.0;
    for (i=0;i<n;i++)
    {
        c[i+1] = thetah[n+i];
    }

    // Call MATLAB polystab function
    mexCallMATLAB(1, &out, 1, &in, "polystab");

    // Alter thetah according to output of polystab
    d = mxGetPr(out);
    for (i=0;i<n;i++)
    {
        thetah[n+i] = d[i+1];
    }

    mxDestroyArray(in);
    mxDestroyArray(out);
}

void polystaba(double *thetah, int n, double *a, double *c)
{
    // Declarations
    mxArray *in, *out;
    double *atmp;
    int i;

    // Create matrix and pointer to it
    in = mxCreateDoubleMatrix(n+1,1,mxREAL);
    atmp = mxGetPr(in);

    // Fill matrix
    atmp[0] = 1.0;
    for (i=0;i<n;i++)
        atmp[i+1] = thetah[i];

    // Call MATLAB polystab function
    mexCallMATLAB(1, &out, 1, &in, "polystab");

    // Alter thetah according to output of polystab
    atmp = mxGetPr(out);
    for (i=0;i<n;i++)
    {
        a[i] = atmp[i+1];
        c[i] = thetah[i+n];
    }

    mxDestroyArray(in);
    mxDestroyArray(out);
}

void mexFunction(
```

```

        int          nlhs,
        mxArray      *plhs[],
        int          nrhs,
        const mxArray *prhs[] )
{
    // Declare function inputs
    double *y = mxGetPr(prhs[0]);
    int n = (int) mxGetScalar(prhs[1]);
    int k = (int) mxGetScalar(prhs[2]);
    double lambda = mxGetScalar(prhs[3]);
    double *thetah = mxGetPr(prhs[4]);
    double *covmat = mxGetPr(prhs[5]);
    double *phim = mxGetPr(prhs[6]);
    double *psim = mxGetPr(prhs[7]);

    // Declare working variables
    int len = mxGetM(prhs[0]);
    int cols = mxGetN(prhs[0]);
    int i, j, h, t, order, d = 2*n;
    double *a, *c, *f, *g, *yhat, *yhatk, *thetam, *tempc;
    double *W, *theta, *phi, *psi, *ftst, *gtst;
    double *epsilon, *num, den=0, tmp, *L, *covmattmp, residual, pred, tempssil,
    temppsi2;
    a = calloc(n, sizeof(double));
    c = calloc(n, sizeof(double));
    f = calloc(k-1, sizeof(double));
    g = calloc(n, sizeof(double));
    yhat = calloc(len, sizeof(double));
    thetam = calloc(d, sizeof(double));
    num = calloc(d, sizeof(double));
    L = calloc(d, sizeof(double));

    // Declare function outputs
    plhs[0]=mxCreateDoubleMatrix(len,cols,mxREAL);
    plhs[1]=mxCreateDoubleMatrix(len,cols,mxREAL);
    plhs[2]=mxCreateDoubleMatrix(d,d,mxREAL);
    plhs[3]=mxCreateDoubleMatrix(d,1,mxREAL);
    plhs[4]=mxCreateDoubleMatrix(d,1,mxREAL);
    plhs[5]=mxCreateDoubleMatrix(d,1,mxREAL);
    yhatk = mxGetPr(plhs[0]);
    epsilon = mxGetPr(plhs[1]);
    covmattmp = mxGetPr(plhs[2]);
    theta = mxGetPr(plhs[3]);
    phi = mxGetPr(plhs[4]);
    psi = mxGetPr(plhs[5]);

    // Initialize coefficients
    for (i=0;i<n;i++)
    {
        a[i] = MRES;
        c[i] = MRES;
        g[i] = MRES;
    }
    for (i=0;i<k-1;i++)
        f[i] = MRES;

    // Create phi arrays
    for (i=0;i<d;i++)
        phi[i] = phim[i];
    for (i=0;i<n;i++)
        phim[i] = phi[i+n];
    for (i=n;i<d;i++)
        phim[i] = phi[i-n] - phi[i];
    for (i=0;i<d;i++)
        theta[i] = thetah[i];

    // Ensure coefficients are stable
    polystabc(theta, n);

    // Loop algorithm
    for (t=0;t<=len-k;t++)
    {
        // Calculate current k-step and 1-step prediction
        for (i=0;i<d;i++)
        {
            yhatk[t+k-1] = yhatk[t+k-1] + phim[i]*thetam[i];
            yhat[t] = yhat[t] + phi[i]*theta[i];

```

```

}

// Calculate error
epsilon[t] = y[t]-yhat[t];

// Calculate prefilter L
den = 0;
for (i=0;i<d;i++)
{
    tmp = 0;
    num[i] = 0;
    for (j=0;j<d;j++)
    {
        num[i] = num[i] + covmat[j*d+i]*psi[j];
        tmp = tmp + psi[j]*covmat[i*d+j];
    }
    den = den + tmp*psi[i];
}
for (i=0;i<d;i++)
{
    L[i] = num[i]/(lambda+den);
}

// Update covariance matrix
for (i=0;i<d;i++)
{
    for (j=0;j<d;j++)
    {
        tmp = 0;
        for (h=0;h<d;h++)
        {
            tmp = tmp + L[j]*psi[h]*covmat[i*d+h];
        }
        covmattmp[i*d+j] = (covmat[i*d+j] - tmp)/lambda;
    }
    for (j=0;j<d;j++)
    {
        covmat[i*d+j] = covmattmp[i*d+j];
    }
}

// Update theta
for (i=0;i<d;i++)
{
    theta[i] = theta[i] + L[i]*epsilon[t];
}

// Ensure filter coefficients are stable
polystabc(theta, n);

// Calculate f & g coefficients
polystaba(theta, n, a, c);
//if (k > 1)
//{
    for (i=0;i<k-1;i++)
    {
        if (i < n)
        {
            order = i;
            f[i] = c[i]-a[i];
        }
        else
        {
            order = n;
            f[i] = 0;
        }

        //f[i] = (i < n ? c[i]-a[i] : 0);
        while (order)
        {
            f[i] = f[i] - a[order-1]*f[i-order];
            order = order - 1;
        }
    }
    for (i=0;i<n;i++)
    {
        if ((n-i) < k)

```

```

        {
            g[i] = 0;
            order = n - i;
        }
        else
        {
            g[i] = c[k+i-1] - a[k+i-1];
            order = k - 1;
        }
        while (order)
        {
            g[i] = g[i] - f[k-order-1]*a[order+i-1];
            order = order - 1;
        }
    }
}

//}
for (i=0;i<n;i++)
{
    thetam[i] = c[i];
    thetam[i+n] = g[i];
}

// Calculate the residual & the prediction
residual = 0;
for (i=0;i<d;i++)
    residual = residual + phi[i]*theta[i];
residual = y[t] - residual;
pred = 0;
for (i=0;i<d;i++)
    pred = pred + phim[i]*thetam[i];

// Update data points and gradient
temppsil = 0;
for (i=0;i<n;i++)
    temppsil = temppsil + (-psi[i])*c[i];
temppsil = -y[t] + temppsil;
temppsi2 = residual;
for (i=0;i<n;i++)
    temppsi2 = temppsi2 - psi[i+n]*c[i];
for (i=0;i<n-1;i++)
{
    phi[n-i-1] = phi[n-i-2];
    phi[d-i-1] = phi[d-i-2];
    psi[n-i-1] = psi[n-i-2];
    psi[d-i-1] = psi[d-i-2];
    phim[n-i-1] = phim[n-i-2];
    phim[d-i-1] = phim[d-i-2];
}
phi[0] = -y[t];
phi[n] = residual;
psi[0] = temppsil;
psi[n] = temppsi2;
phim[0] = -pred;
phim[n] = y[t];
}

// Create output
for (i=0;i<d;i++)
{
    for (j=0;j<d;j++)
        covmattmp[i*d+j] = covmat[i*d+j];
}

// Free dynamic memory
free(a);
free(c);
free(f);
free(g);
free(yhat);
free(thetam);
free(num);
free(L);
}

```

Appendix L

Statistical Predictor

```
function [A,e] = vdkp(y,p,k)

if nargin < 3
    k = 10;
end
if nargin < 2
    p = 50;
end
if k ~= 1 && k ~= 10
    disp('Error - k must be 1 or 10');
    return;
end

len = length(y);
percs = length(p);
[r,c] = size(y);
if c>r
    y = y';
    A = zeros(c,percs);
else
    A = zeros(r,percs);
end

tmpy = zeros(len,1);
z = y;
for n = 1:9
    tmpy(n) = mean(y(1:n));
end
for n = 10:len
    tmpy(n) = mean(y(n-9:n));
end
y = tmpy;

if k == 10
    betal = 6e-4;
    alpha2 = 0.4;
    beta2 = 2e-4;
    gamma2 = 0.43;
else
    betal = 8.7e-4;
    alpha2 = 0.77;
    beta2 = 2.9e-4;
    gamma2 = 0.25;
end

for n = 2*k+1:len
    sigmaA = max([y(n-k) 0])*sqrt(beta2*k)+gamma2*abs(y(n-k)-y(n-2*k));
    mA = y(n-k) + alpha2*(y(n-k)-y(n-2*k));
    A(n) = mA+(2*sigmaA/pi)*asinh(tan(pi*(0.5-(p/100))));
end
e = z-A;
```

Appendix M

ADALINE Predictor

```
#include "mex.h"

void mexFunction(
    int          nlhs,
    mxArray      *plhs[],
    int          nrhs,
    const mxArray *prhs[] )
{
    double *ts = mxGetPr(prhs[0]);
    int n = (int) mxGetScalar(prhs[1]);
    int k = (int) mxGetScalar(prhs[2]);
    double mu = mxGetScalar(prhs[3]);
    double *Wi = mxGetPr(prhs[4]);
    double p1 = mxGetScalar(prhs[5]);

    int len = mxGetM(prhs[0]);
    int cols = mxGetN(prhs[0]);
    double b=0, *W, *Y, *P, *E, err;
    int t, c;
    Y = mxCalloc(n, sizeof(double));

    plhs[0]=mxCreateDoubleMatrix(len,cols,mxREAL);
    plhs[1]=mxCreateDoubleMatrix(len,cols,mxREAL);
    plhs[2]=mxCreateDoubleMatrix(n,1,mxREAL);
    P = mxGetPr(plhs[0]);
    E = mxGetPr(plhs[1]);
    W = mxGetPr(plhs[2]);

    for (t=0;t<k-1;t++)
    {
        P[t] = p1;
    }
    for (t=0;t<n;t++)
    {
        W[t] = Wi[t];
        Y[t] = p1;
    }

    for (t=0;t<len-k+1;t++)
    {
        for(c=0;c<n;c++)
        {
            P[t+k-1] = P[t+k-1] + W[c]*Y[c];
        }
        P[t+k-1] = P[t+k-1] + b;
        E[t] = ts[t] - P[t+k-1];
        err = 2*mu*E[t];
        b = b + err;
        for(c=0;c<n;c++)
        {
            W[c] = W[c] + err*Y[c];
            Y[c] = ((c<n-1) ? Y[c+1] : ts[t]);
        }
    }
    mxFree(Y);
}
```

Appendix N

Hybrid Predictor

```
#include "mex.h"

#define MPRES 2.2204e-16

void polystabc(double *thetah, int n)
{
    // Declarations
    mxArray *in, *out;
    double *c, *d;
    int i;

    // Create matrix and pointer to it
    in = mxCreateDoubleMatrix(n+1,1,mxREAL);
    c = mxGetPr(in);

    // Fill matrix
    c[0] = 1.0;
    for (i=0;i<n;i++)
    {
        c[i+1] = thetah[n+i];
    }

    // Call MATLAB polystab function
    mexCallMATLAB(1, &out, 1, &in, "polystab");

    // Alter thetah according to output of polystab
    d = mxGetPr(out);
    for (i=0;i<n;i++)
    {
        thetah[n+i] = d[i+1];
    }

    mxDestroyArray(in);
    mxDestroyArray(out);
}

void polystaba(double *thetah, int n, double *a, double *c)
{
    // Declarations
    mxArray *in, *out;
    double *atmp;
    int i;

    // Create matrix and pointer to it
    in = mxCreateDoubleMatrix(n+1,1,mxREAL);
    atmp = mxGetPr(in);

    // Fill matrix
    atmp[0] = 1.0;
    for (i=0;i<n;i++)
        atmp[i+1] = thetah[i];

    // Call MATLAB polystab function
    mexCallMATLAB(1, &out, 1, &in, "polystab");

    // Alter thetah according to output of polystab
    atmp = mxGetPr(out);
    for (i=0;i<n;i++)
    {
        a[i] = atmp[i+1];
        c[i] = thetah[i+n];
    }

    mxDestroyArray(in);
    mxDestroyArray(out);
}
```



```

void rtarma(double *yhatk, double *epsilon, double *y, int n, int k, int frq, int len,
double init)
{
    // Creating working variables
    int i, j, h, t, order, d = 2*n;
    double *a, *c, *f, *g, *yhat, *thetam;
    double *theta, *phi, *psi, *phim, *psim;
    double *num, *L, *covmat, *covmattmp;
    double den=0, tmp, residual, pred, tempps1, tempps2, lambda=0.95;
    int stop;
    a = mxCalloc(n, sizeof(double));
    c = mxCalloc(n, sizeof(double));
    f = mxCalloc(k-1, sizeof(double));
    g = mxCalloc(n, sizeof(double));
    yhat = mxCalloc(len, sizeof(double));
    thetam = mxCalloc(d, sizeof(double));
    theta = mxCalloc(d, sizeof(double));
    num = mxCalloc(d, sizeof(double));
    L = mxCalloc(d, sizeof(double));
    phi = mxCalloc(d, sizeof(double));
    psi = mxCalloc(d, sizeof(double));
    phim = mxCalloc(d, sizeof(double));
    psim = mxCalloc(d, sizeof(double));
    covmat = mxCalloc(d*d, sizeof(double));
    covmattmp = mxCalloc(d*d, sizeof(double));
    if (frq==20)
    {
        lambda = 1-0.0003; //0.009;
    }
    else if (frq==40)
    {
        lambda = 1-0.00008; //0.0025;
    }
    else
    {
        lambda = 1-0.00005; //0.0015;
    }

    for (i=0;i<n;i++)
    {
        a[i] = MRES;
        c[i] = MRES;
        g[i] = MRES;
    }
    for (i=0;i<k-1;i++)
        f[i] = MRES;

    // Create phi arrays
    for (i=0;i<d;i++)
    {
        phi[i] = phim[i];
        theta[i] = 0;
        covmat[i*d+i] = 10000;
    }
    for (i=0;i<n;i++)
    {
        phim[i] = phi[i+n];
        phi[i] = init;
    }
    for (i=n;i<d;i++)
        phim[i] = phi[i-n] - phi[i];

    // Ensure coefficients are stable
    polystabc(theta, n);

    // Loop algorithm
    for (t=0;t<=len-k;t++)
    {
        // Calculate current k-step and 1-step prediction
        for (i=0;i<d;i++)
        {
            yhatk[t+k-1] = yhatk[t+k-1] + phim[i]*thetam[i];
            yhat[t] = yhat[t] + phi[i]*theta[i];
        }

        // Calculate error

```

```

epsilon[t] = y[t]-yhat[t];

// Calculate prefilter L
den = 0;
for (i=0;i<d;i++)
{
    tmp = 0;
    num[i] = 0;
    for (j=0;j<d;j++)
    {
        num[i] = num[i] + covmat[j*d+i]*psi[j];
        tmp = tmp + psi[j]*covmat[i*d+j];
    }
    den = den + tmp*psi[i];
}
for (i=0;i<d;i++)
{
    L[i] = num[i]/(lambda+den);
}

// Update covariance matrix
for (i=0;i<d;i++)
{
    for (j=0;j<d;j++)
    {
        tmp = 0;
        for (h=0;h<d;h++)
        {
            tmp = tmp + L[j]*psi[h]*covmat[i*d+h];
        }
        covmattmp[i*d+j] = (covmat[i*d+j] - tmp)/lambda;
    }
    for (j=0;j<d;j++)
    {
        covmat[i*d+j] = covmattmp[i*d+j];
    }
}

// Update theta
for (i=0;i<d;i++)
{
    theta[i] = theta[i] + L[i]*epsilon[t];
}

// Ensure filter coefficients are stable
polystabc(theta, n);

// Calculate f & g coefficients
polystaba(theta, n, a, c);
for (i=0;i<k-1;i++)
{
    if (i < n)
    {
        order = i;
        f[i] = c[i]-a[i];
    }
    else
    {
        order = n;
        f[i] = 0;
    }
    while (order)
    {
        f[i] = f[i] - a[order-1]*f[i-order];
        order = order - 1;
    }
}
for (i=0;i<n;i++)
{
    if ((n-i) < k)
    {
        g[i] = 0;
        order = n - i;
    }
    else
    {
        g[i] = c[k+i-1] - a[k+i-1];
    }
}

```

```

        order = k - 1;
    }
    while (order)
    {
        g[i] = g[i] - f[k-order-1]*a[order+i-1];
        order = order - 1;
    }
}
for (i=0;i<n;i++)
{
    thetam[i] = c[i];
    thetam[i+n] = g[i];
}

// Calculate the residual & the prediction
residual = 0;
for (i=0;i<d;i++)
    residual = residual + phi[i]*theta[i];
residual = y[t] - residual;
pred = 0;
for (i=0;i<d;i++)
    pred = pred + phim[i]*thetam[i];

// Update data points and gradient
temppsil = 0;
for (i=0;i<n;i++)
    temppsil = temppsil + (-psi[i])*c[i];
temppsil = -y[t] + temppsil;
temppsi2 = residual;
for (i=0;i<n;i++)
    temppsi2 = temppsi2 - psi[i+n]*c[i];
for (i=0;i<n-1;i++)
{
    phi[n-i-1] = phi[n-i-2];
    phi[d-i-1] = phi[d-i-2];
    psi[n-i-1] = psi[n-i-2];
    psi[d-i-1] = psi[d-i-2];
    phim[n-i-1] = phim[n-i-2];
    phim[d-i-1] = phim[d-i-2];
}
phi[0] = -y[t];
phi[n] = residual;
psi[0] = temppsil;
psi[n] = temppsi2;
phim[0] = -pred;
phim[n] = y[t];
}
// Free dynamic memory
mxFree(a);
mxFree(c);
mxFree(f);
mxFree(g);
mxFree(yhat);
mxFree(thetam);
mxFree(num);
mxFree(L);
mxFree(phi);
mxFree(psi);
mxFree(phim);
mxFree(psim);
mxFree(covmat);
mxFree(covmattmp);
}

void rtadaline(double *P, double *E, double *ts, int n, int k, int frq, int len,
double init)
{
    int t, c;
    double b=0;
    double mu=0.05/2;
    double err, *W, *Y;
    int stop;
    W = mxCalloc(n, sizeof(double));
    Y = mxCalloc(n, sizeof(double));
    if (frq==20)
    {
        mu = 0.0003; //0.009;

```

```

    }
    else if (frq==40)
    {
        mu = 0.00008; //0.0025;
    }
    else
    {
        mu = 0.00005; //0.0015;
    }
    for (c=0;c<k-1;c++)
    {
        P[c] = init;
    }
    for (c=0;c<n;c++)
    {
        Y[c] = init;
    }
    W[n-1] = 0.4;
    W[n-2] = 0.3;
    W[n-3] = 0.3;

    for (t=0;t<len-k+1;t++)
    {
        for(c=0;c<n;c++)
        {
            P[t+k-1] = P[t+k-1] + W[c]*Y[c];
        }
        P[t+k-1] = P[t+k-1] + b;
        E[t] = ts[t] - P[t+k-1];
        err = 2*mu*E[t];
        b = b + err;
        for(c=0;c<n;c++)
        {
            W[c] = W[c] + err*Y[c];
            Y[c] = ((c<n-1) ? Y[c+1] : ts[t]);
        }
    }
    mxFree(W);
    mxFree(Y);
}

void mexFunction(
    int          nlhs,
    mxArray      *plhs[],
    int          nrhs,
    const mxArray *prhs[] )
{
    // Declare function inputs
    double *y = mxGetPr(prhs[0]);
    int n = (int) mxGetScalar(prhs[1]);
    int k = (int) mxGetScalar(prhs[2]);
    int f = (int) mxGetScalar(prhs[3]);
    double *init = mxGetPr(prhs[4]);
    //double *init = mxGetPr(prhs[4]);

    // Declare working variables
    int rows = mxGetM(prhs[0]);
    int cols = mxGetN(prhs[0]);
    int i,j,ind;
    double *prediction, *Padaline, *Parma, *Eadaline, *Earma;
    double EMarma = 0.0;
    double EMadaline = 0.0;
    Eadaline = mxCalloc(rows*cols, sizeof(double));
    Earma = mxCalloc(rows*cols, sizeof(double));

    // Declare function outputs
    plhs[0] = mxCreateDoubleMatrix(rows,cols,mxREAL);
    plhs[1] = mxCreateDoubleMatrix(rows,cols,mxREAL);
    plhs[2] = mxCreateDoubleMatrix(rows,cols,mxREAL);
    prediction = mxGetPr(plhs[0]);
    Parma = mxGetPr(plhs[1]);
    Padaline = mxGetPr(plhs[2]);

    for (i=0;i<cols;i++)
    {
        rtarma(Parma+(i*rows), Earma+(i*rows), y+(i*rows), n, k, f, rows, init[i]);
        rtadaline(Padaline+(i*rows), Eadaline+(i*rows), y+(i*rows), n, k, f, rows, init[i]);
    }
}

```

```

for (j=0;j<k;j++)
{
    ind = i*cols+j;
    prediction[ind] = init[i];
}
for (j=k;j<rows;j++)
{
    ind = i*cols+j;
    EMarma = 0.92*EMarma + Earma[ind-1]*Earma[ind-1];
    EMadaline = 0.92*EMadaline + Eadaline[ind-1]*Eadaline[ind-1];
    prediction[ind] = ((EMarma<EMadaline) ? Parma[ind] : Padaline[ind]);
}

}

mxFree(Eadaline);
mxFree(Earma);
}

```

Appendix O

Pre-processing Identification Stage

```
% Program to get list of days

clear;

ilength = 300;

day = 1;
month = 9;
year = '97';
endday = 31;
endmonth = 8;
endyear = '98';
simflag = 1;
daysinyear = 0;

dtxt = int2str(day);
if day < 10; dtxt = ['0' dtxt]; end
mtxt = int2str(month);
if month < 10; mtxt = ['0' mtxt]; end
date = [year mtxt dtxt];

no_spikes = 0;
no_drops = 0;
no_gas = 0;
no_20 = 0;
no_40 = 0;
no_50 = 0;
fd_20 = 0;
fd_40 = 0;
fd_50 = 0;
gj_20 = 0;
gj_40 = 0;
gj_50 = 0;
hspike20 = 8;
hspike40 = 25;
hspike50 = 25;
dspike20 = -3.5;
dspike40 = -4.5;
dspike50 = -5;
mgas20 = 1;
mgas40 = 1.5;
mgas50 = 3.5;
startlength20 = 1;
startlength40 = 1;
startlength50 = 1;

while simflag
    daysinyear = daysinyear + 1;
    overlength50 = 0;
    overlength40 = 0;
    overlength20 = 0;
    spike50 = 0;
    spike40 = 0;
    spike20 = 0;
    falsedrop50 = 0;
    falsedrop40 = 0;
    falsedrop20 = 0;
    gasjump50 = 0;
    gasjump40 = 0;
    gasjump20 = 0;
    ex50 = 0;
    ex40 = 0;
    ex20 = 0;
    count20 = 0;
    count40 = 0;
    count50 = 0;
    match20 = 1;
```

```

match40 = 1;
match50 = 1;
if exist(['Data2\' num2month(eval(date(3:4))) date(1:2) \'s50_\' date '.mat'])
    ex50 = 1;
end
if exist(['Data2\' num2month(eval(date(3:4))) date(1:2) \'s40_\' date '.mat'])
    ex40 = 1;
end
if exist(['Data2\' num2month(eval(date(3:4))) date(1:2) \'s20_\' date '.mat'])
    ex20 = 1;
end

if ex50 || ex40 || ex20

    if ex50
        tmpcell = load(['Data2\' num2month(eval(date(3:4))) date(1:2) \'s50_\'
date]);
        eval(['day0_s50 = tmpcell.s50_\' date ';'']);
        eval(['clear tmpcell.s50_\' date ';'']);
        x = day0_s50(:,3);
        gas = day0_s50(:,2)-day0_s50(:,1);
        count50 = 0;
        startflag50 = 1;
        Ar50 = day0_s50(1,1);
        At50 = day0_s50(1,2);
        for c = 1:length(x)
            if x(c) ~= 1 && x(c) ~= 3
                count50 = count50 + 1;
            else
                if startflag50
                    startlength50 = c;
                    startflag50 = 0;
                end
                if (count50 > ilength)
                    overlength50 = 1;
                elseif c-count50-1 > 0
                    if (count50 > ilength)
                        overlength50 = 1;
                    end
                end
                count50 = 0;
            end
            if x(c)==1 || x(c)==3
                if abs(day0_s50(c,1)-Ar50) > hspike50
                    spike50 = 1;
                end
                if abs(day0_s50(c,2)-At50) > hspike50
                    spike50 = 1;
                end
                if gas(c) > mgas50
                    gasjump50 = 1;
                end
                if day0_s50(c,1) < dspike50
                    falsedrop50 = 1;
                end
            end
            Ar50 = day0_s50(c,1);
            At50 = day0_s50(c,2);
        end
        if spike50
            no_50 = no_50 + 1;
        end
        if falsedrop50
            fd_50 = fd_50 + 1;
            disp(['falsedrop 50 ' date ' ' int2str(c)]);
        end
        if gasjump50
            gj_50 = gj_50 + 1;
            disp(['gasjump 50 ' date ' ' int2str(c)]);
        end
        if count50 > ilength
            overlength50 = 1;
        end
    else
        overlength50 = 1;
    end
end

```

```

if ex40
    tmpcell = load(['Data2\' num2month(eval(date(3:4))) date(1:2) '\s40_'
date]);
    eval(['day0_s40 = tmpcell.s40_' date ';'']);
    eval(['clear tmpcell.s40_' date ';'']);
    x = day0_s40(:,3);
    gas = day0_s40(:,2)-day0_s40(:,1);
    count40 = 0;
    startflag40 = 1;
    Ar40 = day0_s40(1,1);
    At40 = day0_s40(1,2);
    for c = 1:length(x)
        if x(c) ~= 1 && x(c) ~= 3
            count40 = count40 + 1;
        else
            if startflag40
                startlength40 = c;
                startflag40 = 0;
            end
            if (count40 > ilength)
                overlength40 = 1;
            elseif c-count40-1 > 0
                if (count40 > ilength)
                    overlength40 = 1;
                end
            end
            count40 = 0;
        end
        if x(c)==1 || x(c)==3
            if abs(day0_s40(c,1)-Ar40) > hspike40
                spike40 = 1;
            end
            if abs(day0_s40(c,2)-At40) > hspike40
                spike40 = 1;
            end
            if gas(c) > mgas40
                gasjump40 = 1;
            end
            if day0_s40(c,1) < dspike40
                falsedrop40 = 1;
            end
        end
        Ar40 = day0_s40(c,1);
        At40 = day0_s40(c,2);
    end
    if spike40
        no_40 = no_40 + 1;
    end
    if falsedrop40
        fd_40 = fd_40 + 1;
        disp(['falsedrop 40 ' date ' ' int2str(c)]);
    end
    if gasjump40
        gj_40 = gj_40 + 1;
        disp(['gasjump 40 ' date ' ' int2str(c)]);
    end
    if count40 > ilength
        overlength40 = 1;
    end
else
    overlength40 = 1;
end

if ex20
    tmpcell = load(['Data2\' num2month(eval(date(3:4))) date(1:2) '\s20_'
date]);
    eval(['day0_s20 = tmpcell.s20_' date ';'']);
    eval(['clear tmpcell.s20_' date ';'']);
    x = day0_s20(:,3);
    gas = day0_s20(:,2)-day0_s20(:,1);
    count20 = 0;
    startflag20 = 1;
    Ar20 = day0_s20(1,1);
    At20 = day0_s20(1,2);
    for c = 1:length(x)
        if x(c) ~= 1 && x(c) ~= 3
            count20 = count20 + 1;

```



```

else
    if startflag20
        startlength20 = c;
        startflag20 = 0;
    end
    if (count20 > ilength)
        overlength20 = 1;
    elseif c-count20-1 > 0
        if (count20 > ilength)
            overlength20 = 1;
        end
    end
    count20 = 0;
end
if x(c)==1 || x(c)==3
    if abs(day0_s20(c,1)-Ar20) > hspike20
        spike20 = 1;
        disp(date);
    end
    if abs(day0_s20(c,2)-At20) > hspike20
        spike20 = 1;
        disp(date);
    end
    if gas(c) > mgas20
        gasjump20 = 1;
    end
    if day0_s20(c,1) < dspike20
        falsedrop20 = 1;
    end
end
Ar20 = day0_s20(c,1);
At20 = day0_s20(c,2);
end
if spike20
    no_20 = no_20 + 1;
end
if falsedrop20
    fd_20 = fd_20 + 1;
    disp(['falsedrop 20 ' date ' ' int2str(c)]);
end
if gasjump20
    gj_20 = gj_20 + 1;
    disp(['gasjump 20 ' date ' ' int2str(c)]);
end
if count20 > ilength
    overlength20 = 1;
end
else
    overlength20 = 1;
end

if count50 > ilength
    if ((ex40 && all(day0_s40(86400-count50+1:86400,3)==1 | day0_s40(86400-
count50+1:86400,3)==3)) || (ex20 && all(day0_s20(86400-count50+1:86400,3)==1 |
day0_s20(86400-count50+1:86400,3)==3)))
        end50 = 1;
    else
        end50 = 0;
    end
end
else
    end50 = 1;
end
if count40 > ilength
    if ((ex50 && all(day0_s50(86400-count40+1:86400,3)==1 | day0_s50(86400-
count40+1:86400,3)==3)) || (ex20 && all(day0_s20(86400-count40+1:86400,3)==1 |
day0_s20(86400-count40+1:86400,3)==3)))
        end40 = 1;
    else
        end40 = 0;
    end
end
else
    end40 = 1;
end
if count20 > ilength
    if ((ex40 && all(day0_s40(86400-count20+1:86400,3)==1 | day0_s40(86400-
count20+1:86400,3)==3)) || (ex50 && all(day0_s50(86400-count20+1:86400,3)==1 |
day0_s50(86400-count20+1:86400,3)==3)))

```

```

        end20 = 1;
    else
        end20 = 0;
    end
else
    end20 = 1;
end

if overlength50 && ex50
    if ex40 && all((day0_s50(startlength50:86400,3)==1 |
day0_s50(startlength50:86400,3)==3) | (day0_s40(startlength50:86400,3)==1 |
day0_s40(startlength50:86400,3)==3));
        match50 = 1;
    elseif ex20 && all((day0_s50(startlength50:86400,3)==1 |
day0_s50(startlength50:86400,3)==3) | (day0_s20(startlength50:86400,3)==1 |
day0_s20(startlength50:86400,3)==3));
        match50 = 1;
    else
        match50 = 0;
    end
end

if overlength40 && ex40
    if ex50 && all((day0_s40(startlength40:86400,3)==1 |
day0_s40(startlength40:86400,3)==3) | (day0_s50(startlength40:86400,3)==1 |
day0_s50(startlength40:86400,3)==3));
        match40 = 1;
    elseif ex20 && all((day0_s40(startlength40:86400,3)==1 |
day0_s40(startlength40:86400,3)==3) | (day0_s20(startlength40:86400,3)==1 |
day0_s20(startlength40:86400,3)==3));
        match40 = 1;
    else
        match40 = 0;
    end
end

if overlength20 && ex20
    if ex40 && all((day0_s20(startlength20:86400,3)==1 |
day0_s20(startlength20:86400,3)==3) | (day0_s40(startlength20:86400,3)==1 |
day0_s40(startlength20:86400,3)==3));
        match20 = 1;
    elseif ex50 && all((day0_s20(startlength20:86400,3)==1 |
day0_s20(startlength20:86400,3)==3) | (day0_s50(startlength20:86400,3)==1 |
day0_s50(startlength20:86400,3)==3));
        match20 = 1;
    else
        match20 = 0;
    end
end

if ~(overlength50 && overlength40 && overlength20) && (end50 && end40 &&
end20) && (match50 && match40 && match20) && ~(falsedrop50 || falsedrop40 ||
falsedrop20) && ~(gasjump50 || gasjump40 || gasjump20) && ~(spike50 || spike40 ||
spike20) && ~strcmp(date,'000105') && ~strcmp(date,'000222')

    daysInfo(simflag).date = date;
    daysInfo(simflag).season = getseason(eval(date(5:6)),eval(date(3:4)));
    simflag = simflag + 1;
end
end

if (eval(date(5:6))==endday) && (eval(date(3:4))==endmonth) &&
strcmp(date(1:2),endyear)
    Ndays = simflag-1;
    simflag = 0;
else
    date = incDate(date);
end
end

Nwinter = 0;
Nspring = 0;
Nsummer = 0;
Nautumn = 0;
for c = 1:length(daysInfo)
    switch daysInfo(c).season
        case 1

```

```
        Nwinter = Nwinter + 1;
    case 2
        Nspring = Nspring + 1;
    case 3
        Nsummer = Nsummer + 1;
    case 4
        Nautumn = Nautumn + 1;
    end
end
save('daysInfo2', 'daysInfo');
```

Appendix P

Pre-processing Interpolation Stage

```
% Program to interpolate data

clear;
load('daysInfo2');
len = length(daysInfo);
date = daysInfo(1).date;
month = num2month(eval(date(3:4)));
ex20 = 0; ex40 = 0; ex50 = 0;
diff2040 = 0.4;
diff4050 = 1.9;
diff2050 = diff2040 + diff4050;
scale2040 = 1.1;
scale4050 = 1.35;
scale2050 = scale2040*scale4050;

% design lpf & hpf
n = 40;
fs = 1;
cutoff = 20e-3;
Wn = cutoff/(fs/2);
blpf = fir1(n,Wn);
bhpfc = fir1(n,Wn,'high');
a = 1;

if exist(['Data2\' num2month(eval(date(3:4))) date(1:2) '\s20_' date '.mat'])
    ex20 = 1;
end
if exist(['Data2\' num2month(eval(date(3:4))) date(1:2) '\s40_' date '.mat'])
    ex40 = 1;
end
if exist(['Data2\' num2month(eval(date(3:4))) date(1:2) '\s50_' date '.mat'])
    ex50 = 1;
end

score = 0;
if ex20
    tst = load(['data2/' month date(1:2) '/S20_' date '.mat']);
    eval(['lastday20 = tst.s20_' date ';' ]);
    count = 0;
    while (lastday20(count+1,3) ~= 1 && lastday20(count+1,3) ~= 3) ||
isnan(lastday20(count+1,1))
        count = count + 1;
    end
    lastday20(1:count,1) = mean(lastday20(count+1:count+10,1));
    lastday20(1:count,2) = mean(lastday20(count+1:count+10,2));
    gaseous20 = lastday20(:,2) - lastday20(:,1);
    gasavg20 = mean(gaseous20);
    [rain20,zr] = filter(blpf,a,lastday20(:,1),lastday20(1,1)*ones(n,1));
    rain20(1:86400-n/2) = rain20(n/2+1:86400);
    rain20(86400-n/2+1:86400) = zr(1:n/2);
    rain20(1:count) = lastday20(1:count,1);
    scint20 = lastday20(:,1) - rain20;
    scint20(1:count) = lastday20(1:count,1);
    score = score + 1;
end
if ex40
    tst = load(['data2/' month date(1:2) '/S40_' date '.mat']);
    eval(['lastday40 = tst.s40_' date ';' ]);
    count = 0;
    while (lastday40(count+1,3) ~= 1 && lastday40(count+1,3) ~= 3) ||
isnan(lastday40(count+1,1))
        count = count + 1;
    end
    lastday40(1:count,1) = mean(lastday40(count+1:count+10,1));
    lastday40(1:count,2) = mean(lastday40(count+1:count+10,2));
    gaseous40 = lastday40(:,2) - lastday40(:,1);
    gasavg40 = mean(gaseous40);
```

```

[rain40,zr] = filter(blpf,a,lastday40(:,1),lastday40(1,1)*ones(n,1));
% [scint40,zs]= filter(bhpf,a,lastday40(:,1),lastday40(1,1)*ones(n,1));
rain40(1:86400-n/2) = rain40(n/2+1:86400);
rain40(86400-n/2+1:86400) = zr(1:n/2);
rain40(1:count) = lastday40(1:count,1);
scint40 = lastday40(:,1) - rain40;
scint40(1:count) = lastday40(1:count,1);
score = score + 2;
end
if ex50
    tst = load(['data2/' month date(1:2) '/S50_' date '.mat']);
    eval(['lastday50 = tst.S50_' date ';'']);
    count = 0;
    while (lastday50(count+1,3) ~= 1 && lastday50(count+1,3) ~= 3) ||
        isnan(lastday50(count+1,1))
        count = count + 1;
    end
    lastday50(1:count,1) = mean(lastday50(count+1:count+10,1));
    lastday50(1:count,2) = mean(lastday50(count+1:count+10,2));
    gaseous50 = lastday50(:,2) - lastday50(:,1);
    gasavg50 = mean(gaseous50);
    [rain50,zr] = filter(blpf,a,lastday50(:,1),lastday50(1,1)*ones(n,1));
    rain50(1:86400-n/2) = rain50(n/2+1:86400);
    rain50(86400-n/2+1:86400) = zr(1:n/2);
    rain50(1:count) = lastday50(1:count,1);
    scint50 = lastday50(:,1) - rain50;
    scint50(1:count) = lastday50(1:count,1);
    score = score + 4;
end

switch score
case 0
    disp('Error, should always be one good attenuation time series');
case 1
    lastday40(:,1) = fscale(20,40,rain20) + sscale(20,40,scint20,1.22,0.61,30);
    lastday40(:,2) = lastday40(:,1) + (gaseous20-gasavg20)*scale2040 + gasavg20 +
diff2040;
    lastday40(:,3) = lastday20(:,3);
    lastday50(:,1) = fscale(20,50,rain20) + sscale(20,50,scint20,1.22,0.61,30);
    lastday50(:,2) = lastday50(:,1) + (gaseous20-gasavg20)*scale2050 + gasavg20 +
diff2050;
    lastday50(:,3) = lastday20(:,3);
case 2
    lastday20(:,1) = fscale(40,20,rain40) + sscale(40,20,scint40,0.61,1.22,30);
    lastday20(:,2) = lastday20(:,1) + (gaseous40-gasavg40)/scale2040 + gasavg40 -
diff2040;
    lastday20(:,3) = lastday40(:,3);
    lastday50(:,1) = fscale(40,50,rain40) + sscale(40,50,scint40,0.61,0.61,30);
    lastday50(:,2) = lastday50(:,1) + (gaseous40-gasavg40)*scale4050 + gasavg40 +
diff4050;
    lastday50(:,3) = lastday40(:,3);
case 3
    if sum(lastday20(:,3)==1|lastday20(:,3)==3) >
sum(lastday40(:,3)==1|lastday40(:,3)==3)
        lastday50(:,1) = fscale(20,50,rain20) +
sscale(20,50,scint20,1.22,0.61,30);
        lastday50(:,2) = lastday50(:,1) + (gaseous20-gasavg20)*scale2050 +
gasavg20 + diff2050;
        lastday50(:,3) = lastday20(:,3);
    else
        lastday50(:,1) = fscale(40,50,rain40) +
sscale(40,50,scint40,0.61,0.61,30);
        lastday50(:,2) = lastday50(:,1) + (gaseous40-gasavg40)*scale4050 +
gasavg40 + diff4050;
        lastday50(:,3) = lastday40(:,3);
    end
case 4
    lastday20(:,1) = fscale(50,20,rain50) + sscale(50,20,scint50,0.61,1.22,30);
    lastday20(:,2) = lastday20(:,1) + (gaseous50-gasavg50)/scale2050 + gasavg50 -
diff2050;
    lastday20(:,3) = lastday50(:,3);
    lastday40(:,1) = fscale(50,40,rain50) + sscale(50,40,scint50,0.61,0.61,30);
    lastday40(:,2) = lastday40(:,1) + (gaseous50-gasavg50)/scale4050 + gasavg50 -
diff4050;
    lastday40(:,3) = lastday50(:,3);
case 5

```

```

        if sum(lastday20(:,3)==1|lastday20(:,3)==3) >
sum(lastday50(:,3)==1|lastday50(:,3)==3)
            lastday40(:,1) = fscale(20,40,rain20) +
sscale(20,40,scint20,1.22,0.61,30);
            lastday40(:,2) = lastday40(:,1) + (gaseous20-gasavg20)*scale2040 +
gasavg20 + diff2040;
            lastday40(:,3) = lastday20(:,3);
        else
            lastday40(:,1) = fscale(50,40,rain50) +
sscale(50,40,scint50,0.61,0.61,30);
            lastday40(:,2) = lastday40(:,1) + (gaseous50-gasavg50)/scale4050 +
gasavg50 - diff4050;
            lastday40(:,3) = lastday50(:,3);
        end
    case 6
        if sum(lastday40(:,3)==1|lastday40(:,3)==3) >
sum(lastday50(:,3)==1|lastday50(:,3)==3)
            lastday20(:,1) = fscale(40,20,rain40) +
sscale(40,20,scint40,0.61,1.22,30);
            lastday20(:,2) = lastday20(:,1) + (gaseous40-gasavg40)/scale2040 +
gasavg40 - diff2040;
            lastday20(:,3) = lastday40(:,3);
        else
            lastday20(:,1) = fscale(50,20,rain50) +
sscale(50,20,scint50,0.61,1.22,30);
            lastday20(:,2) = lastday20(:,1) + (gaseous50-gasavg50)/scale2050 +
gasavg50 - diff2050;
            lastday20(:,3) = lastday50(:,3);
        end
    end

nans20 = isnan(lastday20(:,1));
nans40 = isnan(lastday40(:,1));
nans50 = isnan(lastday50(:,1));
count20 = 0;
count40 = 0;
count50 = 0;
for c = 1:86400
    if nans20(c) || (lastday20(c,3)~=1 && lastday20(c,3)~=3)
        count20 = count20 + 1;
    else
        if count20 > 0
            if all(lastday40(c-count20:c-1,3)==1 | lastday40(c-count20:c-1,3)==3) &&
all(nans40(c-count20:c-1,1)==0) && ex40
                lastday20(c-count20:c-1,1) = fscale(40,20,rain40(c-count20:c-1,1)) +
sscale(40,20,scint40(c-count20:c-1,1),0.61,1.22,30);
                lastday20(c-count20:c-1,2) = lastday20(c-count20:c-1,1) +
(gaseous40(c-count20:c-1)-gasavg40)/scale2040 + gasavg40 - diff2040;
            elseif all(lastday50(c-count20:c-1,3)==1 | lastday50(c-count20:c-1,3)==3)
&& all(nans50(c-count20:c-1,1)==0) && ex50
                lastday20(c-count20:c-1,1) = fscale(50,20,rain50(c-count20:c-1,1)) +
sscale(50,20,scint50(c-count20:c-1,1),0.61,0.61,30);
                lastday20(c-count20:c-1,2) = lastday20(c-count20:c-1,1) +
(gaseous50(c-count20:c-1)-gasavg50)/scale2050 + gasavg50 - diff2050;
            else
                lastday20(c-count20:c-1,1) = linspace(lastday20(c-
count20,1),lastday20(c,1),count20);
                lastday20(c-count20:c-1,2) = linspace(lastday20(c-
count20,2),lastday20(c,2),count20);
            end
            count20 = 0;
        end
    end
    if nans40(c) || (lastday40(c,3)~=1 && lastday40(c,3)~=3)
        count40 = count40 + 1;
    else
        if count40 > 0
            if all(lastday20(c-count40:c-1,3)==1 | lastday20(c-count40:c-1,3)==3) &&
all(nans20(c-count40:c-1,1)==0) && ex20
                lastday40(c-count40:c-1,1) = fscale(20,40,rain20(c-count40:c-1,1)) +
sscale(20,40,scint20(c-count40:c-1,1),1.22,0.61,30);
                lastday40(c-count40:c-1,2) = lastday40(c-count40:c-1,1) +
(gaseous20(c-count40:c-1)-gasavg20)*scale2040 + gasavg20 + diff2040;
            elseif all(lastday50(c-count40:c-1,3)==1 | lastday50(c-count40:c-1,3)==3)
&& all(nans50(c-count40:c-1,1)==0) && ex50
                lastday40(c-count40:c-1,1) = fscale(50,40,rain50(c-count40:c-1,1)) +
sscale(50,40,scint50(c-count40:c-1,1),0.61,0.61,30);

```

```

        lastday40(c-count40:c-1,2) = lastday40(c-count40:c-1,1) +
(gaseous50(c-count40:c-1)-gasavg50)/scale4050 + gasavg50 - diff4050;
    else
        lastday40(c-count40:c-1,1) = linspace(lastday40(c-
count40,1),lastday40(c,1),count40);
        lastday40(c-count40:c-1,2) = linspace(lastday40(c-
count40,2),lastday40(c,2),count40);
    end
    count40 = 0;
end
end
if nans50(c) || (lastday50(c,3)~=1 && lastday50(c,3)~=3)
    count50 = count50 + 1;
else
    if count50 > 0
        if all(lastday20(c-count50:c-1,3)==1 | lastday20(c-count50:c-1,3)==3) &&
all(nans20(c-count50:c-1,1)==0) && ex20
            lastday50(c-count50:c-1,1) = fscale(20,50,rain20(c-count50:c-1,1)) +
sscale(20,50,scint20(c-count50:c-1,1),1.22,0.61,30);
            lastday50(c-count50:c-1,2) = lastday50(c-count50:c-1,1) +
(gaseous20(c-count50:c-1)-gasavg20)*scale2050 + gasavg20 + diff2050;
        elseif all(lastday40(c-count50:c-1,3)==1 | lastday40(c-count50:c-1,3)==3)
&& all(nans40(c-count50:c-1,1)==0) && ex40
            lastday50(c-count50:c-1,1) = fscale(40,50,rain40(c-count50:c-1,1)) +
sscale(40,50,scint40(c-count50:c-1,1),0.61,0.61,30);
            lastday50(c-count50:c-1,2) = lastday50(c-count50:c-1,1) +
(gaseous40(c-count50:c-1)-gasavg40)*scale4050 + gasavg40 + diff4050;
        else
            lastday50(c-count50:c-1,1) = linspace(lastday50(c-
count50,1),lastday50(c,1),count50);
            lastday50(c-count50:c-1,2) = linspace(lastday50(c-
count50,2),lastday50(c,2),count50);
        end
        count50 = 0;
    end
end
end
end

for dayno = 2:len
    date = daysInfo(dayno).date;
    month = num2month(eval(date(3:4)));
    if strcmp(date,'000105');
        stophere = 1;
    end
    lastdate = daysInfo(dayno-1).date;
    if strcmp(date,incDate(lastdate))
        contiguouslast = 1;
    else
        contiguouslast = 0;
    end
    if exist(['Data2\' num2month(eval(date(3:4))) date(1:2) '\s20_' date '.mat'])
        ex20 = 1;
    else
        ex20 = 0;
    end
    if exist(['Data2\' num2month(eval(date(3:4))) date(1:2) '\s40_' date '.mat'])
        ex40 = 1;
    else
        ex40 = 0;
    end
    if exist(['Data2\' num2month(eval(date(3:4))) date(1:2) '\s50_' date '.mat'])
        ex50 = 1;
    else
        ex50 = 0;
    end
end

score = 0;
if ex20
    tst = load(['data2/' month date(1:2) '\s20_' date '.mat']);
    eval(['day20 = tst.s20_' date ';']);
    count20 = 0;
    ind20 = 86400;
    while count20 < 86400 && ((day20(count20+1,3) == 0) ||
isnan(day20(count20+1,1)))
        count20 = count20 + 1;
    end
    while ind20 > 0 && isnan(lastday20(ind20,1))

```

```

        ind20 = ind20 - 1;
    end
    backdist20 = 86400 - ind20 + 1;
    if count20 == 86400
        ex20 = 0;
        score = score - 1;
    end
    if contiguouslast && ex20
        testvec =
linspace(lastday20(ind20,1),day20(count20+1,1),count20+1+backdist20);
        day20(1:count20,1) = testvec(1+backdist20:count20+backdist20);
        lastday20(86401-backdist20:86400,1) = testvec(1:backdist20);
        testvec =
linspace(lastday20(ind20,2),day20(count20+1,2),count20+1+backdist20);
        day20(1:count20,2) = testvec(1+backdist20:count20+backdist20);
        lastday20(86401-backdist20:86400,2) = testvec(1:backdist20);
    elseif ex20
        day20(1:count20,1) = mean(day20(count20+1:count20+10,1));
        day20(1:count20,2) = mean(day20(count20+1:count20+10,2));
        if backdist20 > 1
            lastday20(86401-backdist20:86400,1) = mean(lastday20(86401-backdist20-
10:86401-backdist20,1));
            lastday20(86401-backdist20:86400,2) = mean(lastday20(86401-backdist20-
10:86401-backdist20,2));
        end
    end
    gaseous20 = day20(:,2) - day20(:,1);
    gasavg20 = nanmean(gaseous20);
    [rain20,zr] = filter(blpf,a,day20(:,1),day20(1,1)*ones(n,1));
    rain20(1:86400-n/2) = rain20(n/2+1:86400);
    rain20(86400-n/2+1:86400) = zr(1:n/2);
    rain20(1:count20) = day20(1:count20,1);
    scint20 = day20(:,1) - rain20;
    score = score + 1;
end
if ex40
    tst = load(['data2/' month date(1:2) '/S40_' date '.mat']);
    eval(['day40 = tst.s40_' date ';']);
    count40 = 0;
    ind40 = 86400;
    while count40 < 86400 && ((day40(count40+1,3) == 0) ||
isnan(day40(count40+1,1)))
        count40 = count40 + 1;
    end
    while ind40 > 0 && isnan(lastday40(ind40,1))
        ind40 = ind40 - 1;
    end
    backdist40 = 86400 - ind40 + 1;
    if count40 == 86400
        ex40 = 0;
        score = score - 2;
    end
    if contiguouslast && ex40
        testvec =
linspace(lastday40(ind40,1),day40(count40+1,1),count40+1+backdist40);
        day40(1:count40,1) = testvec(1+backdist40:count40+backdist40);
        lastday40(86401-backdist40:86400,1) = testvec(1:backdist40);
        testvec =
linspace(lastday40(ind40,2),day40(count40+1,2),count40+1+backdist40);
        day40(1:count40,2) = testvec(1+backdist40:count40+backdist40);
        lastday40(86401-backdist40:86400,2) = testvec(1:backdist40);
    elseif ex40
        day40(1:count40,1) = mean(day40(count40+1:count40+10,1));
        day40(1:count40,2) = mean(day40(count40+1:count40+10,2));
        if backdist40 > 1
            lastday40(86401-backdist40:86400,1) = mean(lastday40(86401-backdist40-
10:86401-backdist40,1));
            lastday40(86401-backdist40:86400,2) = mean(lastday40(86401-backdist40-
10:86401-backdist40,2));
        end
    end
    gaseous40 = day40(:,2) - day40(:,1);
    gasavg40 = nanmean(gaseous40);
    [rain40,zr] = filter(blpf,a,day40(:,1),day40(1,1)*ones(n,1));
    rain40(1:86400-n/2) = rain40(n/2+1:86400);
    rain40(86400-n/2+1:86400) = zr(1:n/2);
    rain40(1:count) = day40(1:count,1);

```



```

        scint40 = day40(:,1) - rain40;
        score = score + 2;
    end
    if ex50
        tst = load(['data2/' month date(1:2) '/' s50_ ' date ' '.mat']);
        eval(['day50 = tst.s50_ ' date ' ']);
        count50 = 0;
        ind50 = 86400;
        while count50 < 86400 && ((day50(count50+1,3) == 0) ||
isnan(day50(count50+1,1)))
            count50 = count50 + 1;
        end
        while ind50 > 0 && isnan(lastday50(ind50,1))
            ind50 = ind50 - 1;
        end
        backdist50 = 86400 - ind50 + 1;
        if count50 == 86400
            ex50 = 0;
            score = score - 4;
        end
        if contiguouslast && ex50
            testvec =
linspace(lastday50(ind50,1),day50(count50+1,1),count50+1+backdist50);
            day50(1:count50,1) = testvec(1+backdist50:count50+backdist50);
            lastday50(86401-backdist50:86400,1) = testvec(1:backdist50);
            testvec =
linspace(lastday50(ind50,2),day50(count50+1,2),count50+1+backdist50);
            day50(1:count50,2) = testvec(1+backdist50:count50+backdist50);
            lastday50(86401-backdist50:86400,2) = testvec(1:backdist50);
        elseif ex50
            day50(1:count50,1) = mean(day50(count50+1:count50+10,1));
            day50(1:count50,2) = mean(day50(count50+1:count50+10,2));
            if backdist50 > 1
                lastday50(86401-backdist50:86400,1) = mean(lastday50(86401-backdist50-
10:86401-backdist50,1));
                lastday50(86401-backdist50:86400,2) = mean(lastday50(86401-backdist50-
10:86401-backdist50,2));
            end
        end
        gaseous50 = day50(:,2) - day50(:,1);
        gasavg50 = nanmean(gaseous50);
        [rain50,zr] = filter(blpf,a,day50(:,1),day50(1,1)*ones(n,1));
        rain50(1:86400-n/2) = rain50(n/2+1:86400);
        rain50(86400-n/2+1:86400) = zr(1:n/2);
        rain50(1:count) = day50(1:count,1);
        scint50 = day50(:,1) - rain50;
        score = score + 4;
    end

    switch score
        case 0
            disp('Error, should always be one good attenuation time series');
        case 1
            day40(:,1) = fscale(20,40,rain20) + sscale(20,40,scint20,1.22,0.61,30);
            day40(:,2) = day40(:,1) + (gaseous20-gasavg20)*scale2040 + gasavg20 +
diff2040;
            day40(:,3) = day20(:,3);
            day50(:,1) = fscale(20,50,rain20) + sscale(20,50,scint20,1.22,0.61,30);
            day50(:,2) = day50(:,1) + (gaseous20-gasavg20)*scale2050 + gasavg20 +
diff2050;
            day50(:,3) = day20(:,3);
        case 2
            day20(:,1) = fscale(40,20,rain40) + sscale(40,20,scint40,0.61,1.22,30);
            day20(:,2) = day20(:,1) + (gaseous40-gasavg40)/scale2040 + gasavg40 -
diff2040;
            day20(:,3) = day40(:,3);
            day50(:,1) = fscale(40,50,rain40) + sscale(40,50,scint40,0.61,0.61,30);
            day50(:,2) = day50(:,1) + (gaseous40-gasavg40)*scale4050 + gasavg40 +
diff4050;
            day50(:,3) = day40(:,3);
        case 3
            if sum(day20(:,3)==1|day20(:,3)==3) > sum(day40(:,3)==1|day40(:,3)==3)
                day50(:,1) = fscale(20,50,rain20) +
sscale(20,50,scint20,1.22,0.61,30);
                day50(:,2) = day50(:,1) + (gaseous20-gasavg20)*scale2050 + gasavg20 +
diff2050;
                day50(:,3) = day20(:,3);
            end
        end
    end
end

```

```

else
    day50(:,1) = fscale(40,50,rain40) +
sscale(40,50,scint40,0.61,0.61,30);
    day50(:,2) = day50(:,1) + (gaseous40-gasavg40)*scale4050 + gasavg40 +
diff4050;
    day50(:,3) = day40(:,3);
end
case 4
    day20(:,1) = fscale(50,20,rain50) + sscale(50,20,scint50,0.61,1.22,30);
    day20(:,2) = day20(:,1) + (gaseous50-gasavg50)/scale2050 + gasavg50 -
diff2050;
    day20(:,3) = day50(:,3);
    day40(:,1) = fscale(50,40,rain50) + sscale(50,40,scint50,0.61,0.61,30);
    day40(:,2) = day40(:,1) + (gaseous50-gasavg50)/scale4050 + gasavg50 -
diff4050;
    day40(:,3) = day50(:,3);
case 5
    if sum(day20(:,3)==1|day20(:,3)==3) > sum(day50(:,3)==1|day50(:,3)==3)
        day40(:,1) = fscale(20,40,rain20) +
sscale(20,40,scint20,1.22,0.61,30);
        day40(:,2) = day40(:,1) + (gaseous20-gasavg20)*scale2040 + gasavg20 +
diff2040;
        day40(:,3) = day20(:,3);
    else
        day40(:,1) = fscale(50,40,rain50) +
sscale(50,40,scint50,0.61,0.61,30);
        day40(:,2) = day40(:,1) + (gaseous50-gasavg50)/scale4050 + gasavg50 -
diff4050;
        day40(:,3) = day50(:,3);
    end
case 6
    if sum(day40(:,3)==1|day40(:,3)==3) > sum(day50(:,3)==1|day50(:,3)==3)
        day20(:,1) = fscale(40,20,rain40) +
sscale(40,20,scint40,0.61,1.22,30);
        day20(:,2) = day20(:,1) + (gaseous40-gasavg40)/scale2040 + gasavg40 -
diff2040;
        day20(:,3) = day40(:,3);
    else
        day20(:,1) = fscale(50,20,rain50) +
sscale(50,20,scint50,0.61,1.22,30);
        day20(:,2) = day20(:,1) + (gaseous50-gasavg50)/scale2050 + gasavg50 -
diff2050;
        day20(:,3) = day50(:,3);
    end
end

nans20 = isnan(day20(:,1));
nans40 = isnan(day40(:,1));
nans50 = isnan(day50(:,1));
count20 = 0;
count40 = 0;
count50 = 0;
firsterror20 = 1;
firsterror40 = 1;
firsterror50 = 1;
for c = 1:86400
    if nans20(c) || (day20(c,3)~=1 && day20(c,3)~=3)
        count20 = count20 + 1;
    else
        if count20 > 0
            if all(day40(c-count20:c-1,3)==1 | day40(c-count20:c-1,3)==3) &&
all(nans40(c-count20:c-1,1)==0) && ex40
                day20(c-count20:c-1,1) = fscale(40,20,rain40(c-count20:c-1,1)) +
sscale(40,20,scint40(c-count20:c-1,1),0.61,1.22,30);
                day20(c-count20:c-1,2) = day20(c-count20:c-1,1) + (gaseous40(c-
count20:c-1)-gasavg40)/scale2040 + gasavg40 - diff2040;
            elseif all(day50(c-count20:c-1,3)==1 | day50(c-count20:c-1,3)==3) &&
all(nans50(c-count20:c-1,1)==0) && ex50
                day20(c-count20:c-1,1) = fscale(50,20,rain50(c-count20:c-1,1)) +
sscale(50,20,scint50(c-count20:c-1,1),0.61,0.61,30);
                day20(c-count20:c-1,2) = day20(c-count20:c-1,1) + (gaseous50(c-
count20:c-1)-gasavg50)/scale2050 + gasavg50 - diff2050;
            elseif ~firsterror20
                day20(c-count20:c-1,1) = linspace(day20(c-count20-
1,1),day20(c,1),count20);
                day20(c-count20:c-1,2) = linspace(day20(c-count20-
1,2),day20(c,2),count20);

```

```

elseif sum(day40(c-count20:c-1,3)==1 | day40(c-count20:c-1,3)==3) >
sum(day20(c-count20:c-1,3)==1 | day20(c-count20:c-1,3)==3) && sum(day40(c-count20:c-
1,3)==1 | day40(c-count20:c-1,3)==3) > sum(day50(c-count20:c-1,3)==1 | day50(c-
count20:c-1,3)==3) && ex40
    ind = find(day40(c-count20:c-1,3)==1 | day40(c-count20:c-1,3)==3);
    day20(ind,1) = fscale(40,20,rain40(ind,1)) +
sscale(40,20,scint40(ind,1),0.61,1.22,30);
    day20(ind,2) = day20(ind,1) + (gaseous40(ind)-gasavg40)/scale2040
+ gasavg40 - diff2040;
elseif sum(day50(c-count20:c-1,3)==1 | day50(c-count20:c-1,3)==3) >
sum(day20(c-count20:c-1,3)==1 | day20(c-count20:c-1,3)==3) && ex50
    ind = find(day50(c-count20:c-1,3)==1 | day50(c-count20:c-1,3)==3);
    day20(ind,1) = fscale(50,20,rain50(ind,1)) +
sscale(50,20,scint50(ind,1),0.61,0.61,30);
    day20(ind,2) = day20(ind,1) + (gaseous50(ind)-gasavg50)/scale2050
+ gasavg50 - diff2050;
end
count20 = 0;
end
firsterror20 = 0;
end
if nans40(c) || (day40(c,3)~=1 && day40(c,3)~=3)
    count40 = count40 + 1;
else
    if count40 > 0
        if all(day50(c-count40:c-1,3)==1 | day50(c-count40:c-1,3)==3) &&
all(nans50(c-count40:c-1,1)==0) && ex50
            day40(c-count40:c-1,1) = fscale(50,40,rain50(c-count40:c-1,1)) +
sscale(50,40,scint50(c-count40:c-1,1),0.61,0.61,30);
            day40(c-count40:c-1,2) = day40(c-count40:c-1,1) + (gaseous50(c-
count40:c-1)-gasavg50)/scale4050 + gasavg50 - diff4050;
        elseif all(day20(c-count40:c-1,3)==1 | day20(c-count40:c-1,3)==3) &&
all(nans20(c-count40:c-1,1)==0) && ex20
            day40(c-count40:c-1,1) = fscale(20,40,rain20(c-count40:c-1,1)) +
sscale(20,40,scint20(c-count40:c-1,1),1.22,0.61,30);
            day40(c-count40:c-1,2) = day40(c-count40:c-1,1) + (gaseous20(c-
count40:c-1)-gasavg20)*scale2040 + gasavg20 + diff2040;
        elseif ~firsterror40
            day40(c-count40:c-1,1) = linspace(day40(c-count40-
1,1),day40(c,1),count40);
            day40(c-count40:c-1,2) = linspace(day40(c-count40-
1,2),day40(c,2),count40);
        elseif sum(day50(c-count40:c-1,3)==1 | day50(c-count40:c-1,3)==3) >
sum(day40(c-count40:c-1,3)==1 | day40(c-count40:c-1,3)==3) && sum(day50(c-count40:c-
1,3)==1 | day50(c-count40:c-1,3)==3) > sum(day20(c-count40:c-1,3)==1 | day20(c-
count40:c-1,3)==3) && ex50
            ind = find(day50(c-count40:c-1,3)==1 | day50(c-count40:c-1,3)==3);
            day40(ind,1) = fscale(50,40,rain50(ind,1)) +
sscale(50,40,scint50(ind,1),0.61,0.61,30);
            day40(ind,2) = day40(ind,1) + (gaseous50(ind)-gasavg50)/scale4050
+ gasavg50 - diff4050;
        elseif sum(day20(c-count40:c-1,3)==1 | day20(c-count40:c-1,3)==3) >
sum(day40(c-count40:c-1,3)==1 | day40(c-count40:c-1,3)==3) && ex20
            ind = find(day20(c-count40:c-1,3)==1 | day20(c-count40:c-1,3)==3);
            day40(ind,1) = fscale(20,40,rain20(ind,1)) +
sscale(20,40,scint20(ind,1),1.22,0.61,30);
            day40(ind,2) = day40(ind,1) + (gaseous20(ind)-gasavg20)*scale2040
+ gasavg20 + diff2040;
        end
        count40 = 0;
    end
    firsterror40 = 0;
end
if nans50(c) || (day50(c,3)~=1 && day50(c,3)~=3)
    count50 = count50 + 1;
else
    if count50 > 0
        if all(day40(c-count50:c-1,3)==1 | day40(c-count50:c-1,3)==3) &&
all(nans40(c-count50:c-1,1)==0) && ex40
            day50(c-count50:c-1,1) = fscale(40,50,rain40(c-count50:c-1,1)) +
sscale(40,50,scint40(c-count50:c-1,1),0.61,0.61,30);
            day50(c-count50:c-1,2) = day50(c-count50:c-1,1) + (gaseous40(c-
count50:c-1)-gasavg40)*scale4050 + gasavg40 + diff4050;
        elseif all(day20(c-count50:c-1,3)==1 | day20(c-count50:c-1,3)==3) &&
all(nans20(c-count50:c-1,1)==0) && ex20
            day50(c-count50:c-1,1) = fscale(20,50,rain20(c-count50:c-1,1)) +
sscale(20,50,scint20(c-count50:c-1,1),1.22,0.61,30);

```

```

        day50(c-count50:c-1,2) = day50(c-count50:c-1,1) + (gaseous20(c-
count50:c-1)-gasavg20)*scale2050 + gasavg20 + diff2050;
        elseif ~firsterror50
            day50(c-count50:c-1,1) = linspace(day50(c-count50-
1,1),day50(c,1),count50);
            day50(c-count50:c-1,2) = linspace(day50(c-count50-
1,2),day50(c,2),count50);
            elseif sum(day40(c-count50:c-1,3)==1 | day40(c-count50:c-1,3)==3) >
sum(day50(c-count50:c-1,3)==1 | day50(c-count50:c-1,3)==3) && sum(day40(c-count50:c-
1,3)==1 | day40(c-count50:c-1,3)==3) > sum(day20(c-count50:c-1,3)==1 | day20(c-
count50:c-1,3)==3) && ex40
                ind = find(day40(c-count50:c-1,3)==1 | day40(c-count50:c-1,3)==3);
                day50(ind,1) = fscale(40,50,rain40(ind,1)) +
sscale(40,50,scint40(ind,1),0.61,0.61,30);
                day50(ind,2) = day50(ind,1) + (gaseous40(ind)-gasavg40)*scale4050
+ gasavg40 + diff4050;
                elseif sum(day20(c-count50:c-1,3)==1 | day20(c-count50:c-1,3)==3) >
sum(day50(c-count50:c-1,3)==1 | day50(c-count50:c-1,3)==3) && ex20
                    ind = find(day20(c-count50:c-1,3)==1 | day20(c-count50:c-1,3)==3);
                    day50(ind,1) = fscale(20,50,rain20(ind,1)) +
sscale(20,50,scint20(ind,1),1.22,0.61,30);
                    day50(ind,2) = day50(ind,1) + (gaseous20(ind)-gasavg20)*scale2050
+ gasavg20 + diff2050;
                end
            count50 = 0;
        end
        firsterror50 = 0;
    end
end

c = 86401;
if count20 > 0
    if all(day40(c-count20:c-1,3)==1 | day40(c-count20:c-1,3)==3) && all(nans40(c-
count20:c-1,1)==0) && ex40
        day20(c-count20:c-1,1) = fscale(40,20,rain40(c-count20:c-1,1)) +
sscale(40,20,scint40(c-count20:c-1,1),0.61,1.22,30);
        day20(c-count20:c-1,2) = day20(c-count20:c-1,1) + (gaseous40(c-count20:c-
1)-gasavg40)/scale2040 + gasavg40 - diff2040;
    elseif all(day50(c-count20:c-1,3)==1 | day50(c-count20:c-1,3)==3) &&
all(nans50(c-count20:c-1,1)==0) && ex50
        day20(c-count20:c-1,1) = fscale(50,20,rain50(c-count20:c-1,1)) +
sscale(50,20,scint50(c-count20:c-1,1),0.61,0.61,30);
        day20(c-count20:c-1,2) = day20(c-count20:c-1,1) + (gaseous50(c-count20:c-
1)-gasavg50)/scale2050 + gasavg50 - diff2050;
    else
        day20(c-count20:c-1,1) = NaN;
        day20(c-count20:c-1,2) = NaN;
    end
    count20 = 0;
end
if count40 > 0
    if all(day20(c-count40:c-1,3)==1 | day20(c-count40:c-1,3)==3) && all(nans20(c-
count40:c-1,1)==0) && ex20
        day40(c-count40:c-1,1) = fscale(20,40,rain20(c-count40:c-1,1)) +
sscale(20,40,scint20(c-count40:c-1,1),1.22,0.61,30);
        day40(c-count40:c-1,2) = day40(c-count40:c-1,1) + (gaseous20(c-count40:c-
1)-gasavg20)*scale2040 + gasavg20 + diff2040;
    elseif all(day50(c-count40:c-1,3)==1 | day50(c-count40:c-1,3)==3) &&
all(nans50(c-count40:c-1,1)==0) && ex50
        day40(c-count40:c-1,1) = fscale(50,40,rain50(c-count40:c-1,1)) +
sscale(50,40,scint50(c-count40:c-1,1),0.61,0.61,30);
        day40(c-count40:c-1,2) = day40(c-count40:c-1,1) + (gaseous50(c-count40:c-
1)-gasavg50)/scale4050 + gasavg50 - diff4050;
    else
        day40(c-count40:c-1,1) = NaN;
        day40(c-count40:c-1,2) = NaN;
    end
    count40 = 0;
end
if count50 > 0
    if all(day20(c-count50:c-1,3)==1 | day20(c-count50:c-1,3)==3) && all(nans20(c-
count50:c-1,1)==0) && ex20
        day50(c-count50:c-1,1) = fscale(20,50,rain20(c-count50:c-1,1)) +
sscale(20,50,scint20(c-count50:c-1,1),1.22,0.61,30);
        day50(c-count50:c-1,2) = day50(c-count50:c-1,1) + (gaseous20(c-count50:c-
1)-gasavg20)*scale2050 + gasavg20 + diff2050;

```

```

elseif all(day40(c-count50:c-1,3)==1 | day40(c-count50:c-1,3)==3) &&
all(nans40(c-count50:c-1,1)==0) && ex40
    day50(c-count50:c-1,1) = fscale(40,50,rain40(c-count50:c-1,1)) +
sscale(40,50,scint40(c-count50:c-1,1),0.61,0.61,30);
    day50(c-count50:c-1,2) = day50(c-count50:c-1,1) + (gaseous40(c-count50:c-
1)-gasavg40)*scale4050 + gasavg40 + diff4050;
else
    day50(c-count50:c-1,1) = NaN;
    day50(c-count50:c-1,2) = NaN;
end
count50 = 0;
end

% save day here
eval(['S20_' lastdate ' = lastday20;']);
save(['Data3/S20_' lastdate],['S20_' lastdate]);
eval(['clear S20_' lastdate '']);
eval(['S40_' lastdate ' = lastday40;']);
save(['Data3/S40_' lastdate],['S40_' lastdate]);
eval(['clear S40_' lastdate '']);
eval(['S50_' lastdate ' = lastday50;']);
save(['Data3/S50_' lastdate],['S50_' lastdate]);
eval(['clear S50_' lastdate '']);
disp(['Done ' lastdate ' ' int2str(dayno-1)]);

% increment day
lastday20 = day20;
lastday40 = day40;
lastday50 = day50;
end

% save day here
eval(['S20_' date ' = lastday20;']);
save(['Data3/S20_' date],['S20_' date]);
eval(['clear S20_' date '']);
eval(['S40_' date ' = lastday40;']);
save(['Data3/S40_' date],['S40_' date]);
eval(['clear S40_' date '']);
eval(['S50_' date ' = lastday50;']);
save(['Data3/S50_' date],['S50_' date]);
eval(['clear S50_' date '']);
disp(['Done ' date ' ' int2str(dayno)]);

```

Appendix Q1

Forward control channel link budget at V band

```
clear;
% Input parameters
vsat_diameter = 0.7;           % m
eff_vsat = 0.6;               % ratio
hub_diameter = 2;             % m
eff_hub = 0.68;               % ratio
max_hub_power = 200;          % W
max_vsat_power = 2;           % W
sat_eirp = 51;                % dBW
sat_gt = 8;                   % dB/K
trans_B = 36e6;               % Hz
ulink_freq = 50.42e9;         % Hz
dlink_freq = 39.52e9;         % Hz
ulink_freq_ku = 13.8745e9;
dlink_freq_ku = 11.34525e9;
range = 38.572e6;             % m
c = 3e8;                      % m/s
roll_off = 0.35;              % ratio
bits_per_symbol = 1;          % ratio (bpsk)
Ts_vsat = 10*log10(350);      % dBK
Ts_sat = 10*log10(900);       % dBK
code_rate = '1/3';
EbNo = 2.5;                   % dB (for N=424 qpsk at BER of 1e-6);
uLg = 3;                      % dB
dLg = 1;                      % dB
Leoc_vsat = 3;                % dB
Leoc_hub = 1;                 % dB
Lmisc = 0.5;                  % dB

% Protocol inputs
frame_duration = 20e-3;
guard_time = 5e-6;
no_guards = 2;
preamble_size_ncc_flg = 64*3;
preamble_size_eack = 64*2;
N_ncc_flg = 424;
[M_ncc_flg,block_size_ncc_flg] = getM(N_ncc_flg,code_rate);
N_eak = 48;
[M_eack,block_size_eack] = getM(N_eak,code_rate);
NCC_FLG = block_size_ncc_flg + preamble_size_ncc_flg;
EACK = block_size_eack + preamble_size_eack;
total_bits = NCC_FLG + EACK;
total_duration = frame_duration - no_guards*guard_time;
Rb = total_bits/total_duration;
Rs = Rb/bits_per_symbol;
B = (1+roll_off)*Rs;

% Calculate Lambdas
ulambda = c/ulink_freq;
dlambda = c/dlink_freq;
ulambda_ku = c/ulink_freq_ku;
dlambda_ku = c/dlink_freq_ku;

% Satellite Parameters (Atlantic bird 1)
% Use ku-band information to calculate satellite power since data comes
% from this band. Then re-calculate antenna gains with real G/T
Ts_sat_ku = 10*log10(500);
sGr_ku = sat_gt + Ts_sat_ku;
sGt_ku = sGr_ku - 20*log10(ulink_freq_ku/dlink_freq_ku);
sat_power = 10^((sat_eirp - sGt_ku)/10);
sGr = sGr_ku + 20*log10(ulink_freq/ulink_freq_ku);
sGt = sGt_ku + 20*log10(dlink_freq/dlink_freq_ku);

% Calculate Antenna Gains
Ar_vsat = (pi*vsat_diameter^2)/4;
Ar_hub = (pi*hub_diameter^2)/4;
Ae_vsat = eff_vsat*Ar_vsat;
```

```

Ae_hub = eff_hub*Ar_hub;
uG_vsatsat = 10*log10(4*pi*Ae_vsatsat/(ulambda^2));
dG_vsatsat = 10*log10(4*pi*Ae_vsatsat/(dlambda^2));
uG_hubsat = 10*log10(4*pi*Ae_hubsat/(ulambda^2));
dG_hubsat = 10*log10(4*pi*Ae_hubsat/(dlambda^2));

% Calculate Free Space Path Loss
uLp = 20*log10(4*pi*range/ulambda);
dLp = 20*log10(4*pi*range/dlambda);

% Noise Power Calculation
Np_sat = -228.6 + 10*log10(Rs) + Ts_sat;
Np_vsatsat = -228.6 + 10*log10(Rs) + Ts_vsatsat;

% Calculate C/N
CtoN_req = EbNo + 10*log10(Rb) - 10*log10(Rs);

% Calculate hub uplink power budget
% Assume 1/200 hub power is for control channel
power_ratio = 1/200;
Pr_satsat = 10*log10(max_hub_power*power_ratio) + uG_hubsat + sGr - uLp - Leoc_hubsat - uLg - Lmisc;

% Calculate outbound uplink C/N
CtoN_up = Pr_satsat - Np_satsat;

% Calculate transponder downlink power budget
% Assume transponder power is -occupied bandwidth for outbound control channel
Pr_vsatsat = 10*log10(sat_power*B/trans_B) + dG_vsatsat + sGt - dLp - Leoc_vsatsat - dLg - Lmisc;

% Calculate outbound downlink C/N
CtoN_down = Pr_vsatsat - Np_vsatsat;

% Calculate overall outbound C/N
CtoN_lin = 1/((1/(10^(CtoN_down/10))) + (1/(10^(CtoN_up/10))));
CtoN_tot = 10*log10(CtoN_lin);
CtoN_min_up = 10*log10(1/((1/(10^(CtoN_req/10))) - (1/(10^(CtoN_down/10))));
CtoN_min_down = 10*log10(1/((1/(10^(CtoN_req/10))) - (1/(10^(CtoN_up/10))));

% Link margin
Link_margin_tot = CtoN_tot - CtoN_req;
Link_margin_up = CtoN_up - CtoN_min_up;
Link_margin_down = CtoN_down - CtoN_min_down;

Taca = 275*(1-10^(-(dLg/10)));
no_iterations = 1000;
for iter = 1:no_iterations
    Arain(iter) = iter*Link_margin_down/no_iterations;
    Tarain = 275*(1-10^(-(dLg+Arain(iter))/10));
    dTsys = Tarain-Taca;
    Tsyrain = 10^(Ts_vsatsat/10)+dTsys;
    dNrainsat = 10*log10(Tsyrainsat/(10^(Ts_vsatsat/10)));
    Atotals(iter) = Arain(iter) + dNrainsat;
end
Link_margin_down_effective = interp1(Atotals,Arain,Link_margin_down);

```


Appendix Q2

Return control channel link budget at V band

```
clear;
% Input parameters
vsat_diameter = 0.7; % m
eff_vsat = 0.6; % ratio
hub_diameter = 2; % m
eff_hub = 0.68; % ratio
max_hub_power = 200; % W
max_vsat_power = 2; % W
sat_eirp = 51; % dBW
sat_gt = 8; % dB/K
trans_B = 36e6; % Hz
ulink_freq = 50.42e9; % Hz
dlink_freq = 39.52e9; % Hz
ulink_freq_ku = 13.8745e9;
dlink_freq_ku = 11.34525e9;
range = 38.5e6; % m
c = 3e8; % m/s
roll_off = 0.35; % ratio
bits_per_symbol = 1; % ratio (bpsk)
Ts_hub = 10*log10(350); % dBK
Ts_sat = 10*log10(900); % dBK
code_rate = '1/3';
k = 7;
EbNo = 4.2; % dB (for bpsk at BER of 1e-6 [convolutional!]);
uLg = 3; % dB
dLg = 1; % dB
Leoc_vsat = 3; % dB
Leoc_hub = 1; % dB
Lmisc = 0.5; % dB

% Protocol inputs
frame_duration = 20e-3;
guard_time = 5e-6;
no_stations_per_frame = 8;
no_guards = no_stations_per_frame + 2;
preamble_size_rbic = 64*2;
preamble_size_rcst = 64*2;
preamble_size_ecs = 64*2;
control_bits_rbic = 8 + 8;
control_bits_rcst = 4 + 8 + 8;
control_bits_ecs = 4 + 48 + 8;
block_size_rbic = preamble_size_rbic + (control_bits_rbic + k)/eval(code_rate);
block_size_rcst = no_stations_per_frame*(preamble_size_rcst + (control_bits_rcst + k)/eval(code_rate));
block_size_ecs = preamble_size_ecs + (control_bits_ecs + k)/eval(code_rate);
no_trf_channels = 8;
total_bits = block_size_rbic + block_size_rcst + block_size_ecs;
total_duration = frame_duration - no_guards*guard_time;
Rb = total_bits/total_duration;
Rs = Rb/bits_per_symbol;
B = (1+roll_off)*Rs;

% Calculate Lambdas
ulambda = c/ulink_freq;
dlambda = c/dlink_freq;
ulambda_ku = c/ulink_freq_ku;
dlambda_ku = c/dlink_freq_ku;

% Satellite Parameters (Atlantic bird 1)
% Use ku-band information to calculate satellite power since data comes
% from this band. Then re-calculate antenna gains with real G/T
Ts_sat_ku = 10*log10(500);
sGr_ku = sat_gt + Ts_sat_ku;
sGt_ku = sGr_ku - 20*log10(ulink_freq_ku/dlink_freq_ku);
sat_power = 10^((sat_eirp - sGt_ku)/10);
sGr = sGr_ku + 20*log10(ulink_freq/ulink_freq_ku);
sGt = sGt_ku + 20*log10(dlink_freq/dlink_freq_ku);
```



```

% Calculate Antenna Gains
Ar_vsat = (pi*vsat_diameter^2)/4;
Ar_hub = (pi*hub_diameter^2)/4;
Ae_vsat = eff_vsat*Ar_vsat;
Ae_hub = eff_hub*Ar_hub;
uG_vsat = 10*log10(4*pi*Ae_vsat/ulambda^2);
dG_vsat = 10*log10(4*pi*Ae_vsat/dlambda^2);
uG_hub = 10*log10(4*pi*Ae_hub/ulambda^2);
dG_hub = 10*log10(4*pi*Ae_hub/dlambda^2);

% Calculate Free Space Path Loss
uLp = 20*log10(4*pi*range/ulambda);
dLp = 20*log10(4*pi*range/dlambda);

% Noise Power Calculation
Np_sat = -228.6 + 10*log10(Rs) + Ts_sat;
Np_hub = -228.6 + 10*log10(Rs) + Ts_hub;

% Calculate C/N
CtoN_req = EbNo + 10*log10(Rb) - 10*log10(Rs);

% Calculate vsat uplink power budget
% Assume 4/20 vsat power is for control channel
Pr_sat = 10*log10(max_vsat_power*4/20) + uG_vsat + sGr - uLp - Leoc_vsat - uLg - Lmisc;

% Calculate inbound uplink C/N
CtoN_up = Pr_sat - Np_sat;

% Calculate transponder downlink power budget
% Assume transponder power is ~occupied bandwidth for inbound control channel
Pr_hub = 10*log10(sat_power*B/trans_B) + dG_hub + sGt - dLp - Leoc_hub - dLg - Lmisc;

% Calculate inbound downlink C/N
CtoN_down = Pr_hub - Np_hub;

% Calculate overall inbound C/N
CtoN_lin = 1/((1/(10^(CtoN_down/10))) + (1/(10^(CtoN_up/10))));
CtoN_tot = 10*log10(CtoN_lin);
CtoN_min_up = 10*log10(1/((1/(10^(CtoN_req/10))) - (1/(10^(CtoN_down/10)))));
CtoN_min_down = 10*log10(1/((1/(10^(CtoN_req/10))) - (1/(10^(CtoN_up/10)))));

% Link margin
Link_margin_tot = CtoN_tot - CtoN_req;
Link_margin_up = CtoN_up - CtoN_min_up;
Link_margin_down = CtoN_down - CtoN_min_down;

Taca = 275*(1-10^(-(dLg/10)));
no_iterations = 1000;
for iter = 1:no_iterations
    Arain(iter) = iter*Link_margin_down/no_iterations;
    Tarain = 275*(1-10^(-(dLg+Arain(iter))/10));
    dTsys = Tarain-Taca;
    Tsysrain = 10^(Ts_hub/10)+dTsys;
    dNrain = 10*log10(Tsysrain/(10^(Ts_hub/10)));
    Atotals(iter) = Arain(iter) + dNrain;
end
Link_margin_down_effective = interp1(Atotals,Arain,Link_margin_down);

```

Appendix Q3

Forward control channel link budget at Ku band

```
clear;
% Input parameters
vsat_diameter = 0.8;           % m
eff_vsat = 0.6;               % ratio
hub_diameter = 2;             % m
eff_hub = 0.68;               % ratio
max_hub_power = 200;          % W
max_vsat_power = 2;           % W
sat_eirp = 51;                 % dBW
sat_gt = 8;                   % dB/K
trans_B = 36e6;               % Hz
ulink_freq_ku = 13.8745e9;     % Hz
dlink_freq_ku = 11.34525e9;    % Hz
range = 38.572e6;             % m
c = 3e8;                      % m/s
roll_off = 0.35;              % ratio
bits_per_symbol = 1;          % ratio (bpsk)
Ts_vsat_ku = 10*log10(150);    % dBK
Ts_sat_ku = 10*log10(500);     % dBK
code_rate = '1/2';
EbNo = 2.6;                   % dB (for N=424 qpsk at BER of 1e-6);
uLg = 0.7;                    % dB
dLg = 0.5;                    % dB
Leoc_vsat = 3;                % dB
Leoc_hub = 1;                 % dB
Lmisc = 0.5;                  % dB

% Protocol inputs
frame_duration = 20e-3;
guard_time = 5e-6;
no_guards = 2;
preamble_size_ncc_flc = 64*3;
preamble_size_eack = 64*2;
N_ncc_flc = 424;
[M_ncc_flc, block_size_ncc_flc] = getM(N_ncc_flc, code_rate);
N_eak = 48;
[M_eack, block_size_eack] = getM(N_eak, code_rate);
NCC_FLC = block_size_ncc_flc + preamble_size_ncc_flc;
EACK = block_size_eack + preamble_size_eack;
total_bits = NCC_FLC + EACK;
total_duration = frame_duration - no_guards*guard_time;
Rb = total_bits/total_duration;
Rs = Rb/bits_per_symbol;
B = (1+roll_off)*Rs;

% Calculate Lambdas
ulambda_ku = c/ulink_freq_ku;
dlambda_ku = c/dlink_freq_ku;

% Satellite Parameters (Atlantic bird 1)
% Use ku-band information to calculate satellite power since data comes
% from this band. Then re-calculate antenna gains with real G/T
sGr_ku = sat_gt + Ts_sat_ku;
sGt_ku = sGr_ku - 20*log10(ulink_freq_ku/dlink_freq_ku);
sat_power = 10^((sat_eirp - sGt_ku)/10);

% Calculate Antenna Gains
Ar_vsat = (pi*vsat_diameter^2)/4;
Ar_hub = (pi*hub_diameter^2)/4;
Ae_vsat = eff_vsat*Ar_vsat;
Ae_hub = eff_hub*Ar_hub;
uG_vsat = 10*log10(4*pi*Ae_vsat/(ulambda_ku^2));
dG_vsat = 10*log10(4*pi*Ae_vsat/(dlambda_ku^2));
uG_hub = 10*log10(4*pi*Ae_hub/(ulambda_ku^2));
dG_hub = 10*log10(4*pi*Ae_hub/(dlambda_ku^2));

% Calculate Free Space Path Loss
```

```

uLp_ku = 20*log10(4*pi*range/ulambda_ku);
dLp_ku = 20*log10(4*pi*range/dlambda_ku);

% Noise Power Calculation
Np_sat_ku = -228.6 + 10*log10(Rs) + Ts_sat_ku;
Np_vsats_ku = -228.6 + 10*log10(Rs) + Ts_vsats_ku;

% Calculate C/N
CtoN_req_ku = EbNo + 10*log10(Rb) - 10*log10(Rs);

% Calculate hub uplink power budget
% Assume 1/200 hub power is for control channel
power_ratio = 1/200;
Pr_sat_ku = 10*log10(max_hub_power*power_ratio) + uG_hub + sGr_ku - uLp_ku - Leoc_hub - uLg - Lmisc;

% Calculate outbound uplink C/N
CtoN_up_ku = Pr_sat_ku - Np_sat_ku;

% Calculate transponder downlink power budget
% Assume transponder power is ~occupied bandwidth for outbound control channel
Pr_vsats_ku = 10*log10(sat_power*B/trans_B) + dG_vsats + sGt_ku - dLp_ku - Leoc_vsats - dLg - Lmisc;

% Calculate outbound downlink C/N
CtoN_down_ku = Pr_vsats_ku - Np_vsats_ku;

% Calculate overall outbound C/N
CtoN_lin_ku = 1/((1/(10^(CtoN_down_ku/10))) + (1/(10^(CtoN_up_ku/10))));
CtoN_tot_ku = 10*log10(CtoN_lin_ku);
CtoN_min_up_ku = 10*log10(1/((1/(10^(CtoN_req_ku/10))) - (1/(10^(CtoN_down_ku/10)))));
CtoN_min_down_ku = 10*log10(1/((1/(10^(CtoN_req_ku/10))) - (1/(10^(CtoN_up_ku/10)))));

% Link margin
Link_margin_tot_ku = CtoN_tot_ku - CtoN_req_ku;
Link_margin_up_ku = CtoN_up_ku - CtoN_min_up_ku;
Link_margin_down_ku = CtoN_down_ku - CtoN_min_down_ku;

Taca_ku = 275*(1-10^(-(dLg/10)));
no_iterations = 1000;
Arain_ku = zeros(no_iterations,1);
Atotals_ku = zeros(no_iterations,1);
for iter = 1:no_iterations
    Arain_ku(iter) = iter*Link_margin_down_ku/no_iterations;
    Tarain_ku = 275*(1-10^(-(dLg+Arain_ku(iter))/10)));
    dTsys_ku = Tarain_ku-Taca_ku;
    Tsysrain_ku = 10^(Ts_vsats_ku/10)+dTsys_ku;
    dNrain_ku = 10*log10(Tsysrain_ku/(10^(Ts_vsats_ku/10)));
    Atotals_ku(iter) = Arain_ku(iter) + dNrain_ku;
end
Link_margin_down_effective_ku = interp1(Atotals_ku,Arain_ku,Link_margin_down_ku);

```

Appendix Q4

Return control channel link budget at *Ku* band

```
clear;
% Input parameters
vsat_diameter = 0.8; % m
eff_vsat = 0.6; % ratio
hub_diameter = 2; % m
eff_hub = 0.68; % ratio
max_hub_power = 200; % W
max_vsat_power = 2; % W
sat_eirp = 51; % dBW
sat_gt = 8; % dB/K
trans_B = 36e6; % Hz
ulink_freq_ku = 13.8745e9; % Hz
dlink_freq_ku = 11.34525e9; % Hz
range = 38.572e6; % m
c = 3e8; % m/s
roll_off = 0.35; % ratio
bits_per_symbol = 1; % ratio (bpsk)
Ts_vsat_ku = 10*log10(150); % dBK
Ts_sat_ku = 10*log10(500); % dBK
code_rate = '1/2';
EbNo = 2.6; % dB (for N=424 qpsk at BER of 1e-6);
uLg = 0.7; % dB
dLg = 0.5; % dB
Leoc_vsat = 3; % dB
Leoc_hub = 1; % dB
Lmisc = 0.5; % dB

% Protocol inputs
frame_duration = 20e-3;
guard_time = 5e-6;
no_guards = 2;
preamble_size_ncc_flg = 64*3;
preamble_size_eack = 64*2;
N_ncc_flg = 424;
[M_ncc_flg, block_size_ncc_flg] = getM(N_ncc_flg, code_rate);
N_eak = 48;
[M_eak, block_size_eack] = getM(N_eak, code_rate);
NCC_FLG = block_size_ncc_flg + preamble_size_ncc_flg;
EACK = block_size_eack + preamble_size_eack;
total_bits = NCC_FLG + EACK;
total_duration = frame_duration - no_guards*guard_time;
Rb = total_bits/total_duration;
Rs = Rb/bits_per_symbol;
B = (1+roll_off)*Rs;

% Calculate Lambdas
ulambda_ku = c/ulink_freq_ku;
dlambda_ku = c/dlink_freq_ku;

% Satellite Parameters (Atlantic bird 1)
% Use ku-band information to calculate satellite power since data comes
% from this band. Then re-calculate antenna gains with real G/T
sGr_ku = sat_gt + Ts_sat_ku;
sGt_ku = sGr_ku - 20*log10(ulink_freq_ku/dlink_freq_ku);
sat_power = 10^((sat_eirp - sGt_ku)/10);

% Calculate Antenna Gains
Ar_vsat = (pi*vsat_diameter^2)/4;
Ar_hub = (pi*hub_diameter^2)/4;
Ae_vsat = eff_vsat*Ar_vsat;
Ae_hub = eff_hub*Ar_hub;
uG_vsat = 10*log10(4*pi*Ae_vsat/(ulambda_ku^2));
dG_vsat = 10*log10(4*pi*Ae_vsat/(dlambda_ku^2));
uG_hub = 10*log10(4*pi*Ae_hub/(ulambda_ku^2));
dG_hub = 10*log10(4*pi*Ae_hub/(dlambda_ku^2));

% Calculate Free Space Path Loss
```

```

uLp_ku = 20*log10(4*pi*range/ulambda_ku);
dLp_ku = 20*log10(4*pi*range/dlambda_ku);

% Noise Power Calculation
Np_sat_ku = -228.6 + 10*log10(Rs) + Ts_sat_ku;
Np_vsats_ku = -228.6 + 10*log10(Rs) + Ts_vsats_ku;

% Calculate C/N
CtoN_req_ku = EbNo + 10*log10(Rb) - 10*log10(Rs);

% Calculate hub uplink power budget
% Assume 1/200 hub power is for control channel
power_ratio = 1/200;
Pr_sat_ku = 10*log10(max_hub_power*power_ratio) + uG_hub + sGr_ku - uLp_ku - Leoc_hub
- uLg - Lmisc;

% Calculate outbound uplink C/N
CtoN_up_ku = Pr_sat_ku - Np_sat_ku;

% Calculate transponder downlink power budget
% Assume transponder power is -occupied bandwidth for outbound control channel
Pr_vsats_ku = 10*log10(sat_power*B/trans_B) + dG_vsats + sGt_ku - dLp_ku - Leoc_vsats -
dLg - Lmisc;

% Calculate outbound downlink C/N
CtoN_down_ku = Pr_vsats_ku - Np_vsats_ku;

% Calculate overall outbound C/N
CtoN_lin_ku = 1/((1/(10^(CtoN_down_ku/10))) + (1/(10^(CtoN_up_ku/10))));
CtoN_tot_ku = 10*log10(CtoN_lin_ku);
CtoN_min_up_ku = 10*log10(1/((1/(10^(CtoN_req_ku/10))) - (1/(10^(CtoN_down_ku/10)))));
CtoN_min_down_ku = 10*log10(1/((1/(10^(CtoN_req_ku/10))) - (1/(10^(CtoN_up_ku/10)))));

% Link margin
Link_margin_tot_ku = CtoN_tot_ku - CtoN_req_ku;
Link_margin_up_ku = CtoN_up_ku - CtoN_min_up_ku;
Link_margin_down_ku = CtoN_down_ku - CtoN_min_down_ku;

Taca_ku = 275*(1-10^(-(dLg/10)));
no_iterations = 1000;
Arain_ku = zeros(no_iterations,1);
Atotals_ku = zeros(no_iterations,1);
for iter = 1:no_iterations
    Arain_ku(iter) = iter*Link_margin_down_ku/no_iterations;
    Tarain_ku = 275*(1-10^(-(dLg+Arain_ku(iter))/10)));
    dTsys_ku = Tarain_ku-Taca_ku;
    Tsyrain_ku = 10^(Ts_vsats_ku/10)+dTsys_ku;
    dNrain_ku = 10*log10(Tsyrain_ku/(10^(Ts_vsats_ku/10)));
    Atotals_ku(iter) = Arain_ku(iter) + dNrain_ku;
end
Link_margin_down_effective_ku = interp1(Atotals_ku,Arain_ku,Link_margin_down_ku);

```

Appendix R

Statistical analysis at V band with signal processing only

```
clear;
load('daysInfo2');
len = length(daysInfo);
xo = 0:1e-4:50.5e-3;
xi = 0:1e-4:990.7e-3;
usageo = zeros(length(xo),1);
usagei = zeros(length(xi),1);
currenttime = zeros(len,1);
overflow = 0;
sampletot = 0;
totaltime = 0;
Nstations = 58;
CtoNo_tot = zeros(86400,Nstations);
CtoNi_tot = zeros(86400,Nstations);
outageso = zeros(Nstations,1);
outagesi = zeros(Nstations,1);
outageso_nom = zeros(Nstations,1);
outagesi_nom = zeros(Nstations,1);
outages_nom = zeros(Nstations,1);
outages = zeros(Nstations,1);
results = zeros(10,1);
ts40 = zeros(86400,Nstations+1);
ts50 = zeros(86400,Nstations+1);
gas40 = zeros(86400,Nstations+1);
gas50 = zeros(86400,Nstations+1);

% initialize matrix
for c = 1:Nstations+1
    date = daysInfo(c).date;
    dummycell = load(['Data3/S40_' date '.mat']);
    eval(['ts40(:,c) = dummycell.S40_' date '(:,2);']);
    eval(['gas40(:,c) = dummycell.S40_' date '(:,2) - dummycell.S40_' date '(:,1);']);
    dummycell = load(['Data3/S50_' date '.mat']);
    eval(['ts50(:,c) = dummycell.S50_' date '(:,2);']);
    eval(['gas50(:,c) = dummycell.S50_' date '(:,2) - dummycell.S50_' date '(:,1);']);
end

for m = 1:len
    tic;
    % Change single site time series to joint site time series
    timeshift20 = 14.5*60;
    timeshift40 = 14.5*60;
    timeshift50 = 14.5*60;
    for c40 = 1:86400-timeshift40
        ts40(c40,1) = min(ts40(c40,1),ts40(c40+timeshift40,1));
    end
    for c40 = 86400-timeshift40+1:86400
        ts40(c40,1) = min(ts40(c40,1),ts40(c40-86400+timeshift40,2));
    end
    for c50 = 1:86400-timeshift50
        ts50(c50,1) = min(ts50(c50,1),ts50(c50+timeshift50,1));
    end
    for c50 = 86400-timeshift50+1:86400
        ts50(c50,1) = min(ts50(c50,1),ts50(c50-86400+timeshift50,2));
    end
    end

    load('OutboundTraffic');
    CtoN_up1 = CtoN_up;
    CtoN_down1 = CtoN_down;
    CtoN_tot1 = CtoN_tot;
    CtoN_req1 = CtoN_req;
    Tacal = Tacal;
    load('InboundTraffic');
    CtoN_up2 = CtoN_up;
    CtoN_down2 = CtoN_down;
    CtoN_tot2 = CtoN_tot;
    CtoN_req2 = CtoN_req;
```

```

Taca2 = Taca;
% Calculate Outbound Uplink C/N
CtoNo_u = CtoN_up1 - ts50(:,1);
Tsyscal = 10^(Ts_vsats/10);
CtoNo_u_lin = 10.^(CtoNo_u/10);
for c = 1:Nstations
    Tarain1 = 275*(1-10.^(-(gas40(:,c+1)+ts40(:,c+1))/10));
    dTsys1 = Tarain1-Taca1;
    Tsysrain1 = Tsyscal+dTsys1;
    dNrain1 = 10*log10(Tsysrain1/Tsyscal);
    CtoNo_d = CtoN_down1 - ts40(:,c+1) - dNrain1;
    CtoNo_lin = 1./((1./(10.^(CtoNo_d/10))) + (1./CtoNo_u_lin));
    CtoNo_tot(:,c) = 10*log10(CtoNo_lin);
end
% Calculate Inbound Downlink C/N
Tsysca2 = 10^(Ts_hub/10);
Tarain2 = 275*(1-10.^(-(gas40(:,1)+ts40(:,1))/10));
dTsys2 = Tarain2-Taca2;
Tsysrain2 = Tsysca2+dTsys2;
dNrain2 = 10*log10(Tsysrain2/Tsysca2);
CtoNi_d = CtoN_down2 - ts40(:,1) - dNrain2;
% Calculate Outbound Uplink & Total C/N
CtoNi_d_lin = 10.^(CtoNi_d/10);
for c = 1:Nstations
    CtoNi_u = CtoN_up - ts50(:,c+1);
    CtoNi_lin = 1./((1./CtoNi_d_lin) + (1./(10.^(CtoNi_u/10))));
    CtoNi_tot(:,c) = 10*log10(CtoNi_lin);
end

% Gather statistics
overflowtmpo = zeros(86400,1);
overflowtmpi = zeros(86400,1);
durationso = zeros(86400,1);
durationsi = zeros(86400,1);
for s = 1:86400
    outagestmpo = zeros(Nstations,1);
    outagestmpo_nom = zeros(Nstations,1);
    outagestmpi = zeros(Nstations,1);
    outagestmpi_nom = zeros(Nstations,1);
    tot_duro = 0;
    tot_duri = 0;
    for c = 1:Nstations
        if CtoNo_tot(s,c) < CtoN_tot1 - marginsout(end);
            outagestmpo(c) = 1;
            outageso(c) = outageso(c) + 1;
        end
        if CtoNo_tot(s,c) < CtoN_tot1 - marginsout(1);
            outagestmpo_nom(c) = 1;
            outageso_nom(c) = outageso_nom(c) + 1;
        end
        if CtoNo_tot(s,c) > CtoN_tot1 - marginsout(1)
            tot_duro = tot_duro + durout(1);
        elseif CtoNo_tot(s,c) > CtoN_tot1 - marginsout(2)
            tot_duro = tot_duro + durout(2);
        elseif CtoNo_tot(s,c) > CtoN_tot1 - marginsout(3)
            tot_duro = tot_duro + durout(3);
        elseif CtoNo_tot(s,c) > CtoN_tot1 - marginsout(4)
            tot_duro = tot_duro + durout(4);
        elseif CtoNo_tot(s,c) > CtoN_tot1 - marginsout(5)
            tot_duro = tot_duro + durout(5);
        elseif CtoNo_tot(s,c) > CtoN_tot1 - marginsout(6)
            tot_duro = tot_duro + durout(6);
        elseif CtoNo_tot(s,c) > CtoN_tot1 - marginsout(7)
            tot_duro = tot_duro + durout(7);
        elseif CtoNo_tot(s,c) > CtoN_tot1 - marginsout(8)
            tot_duro = tot_duro + durout(8);
        elseif CtoNo_tot(s,c) > CtoN_tot1 - marginsout(9)
            tot_duro = tot_duro + durout(9);
        end
        if CtoNi_tot(s,c) < CtoN_tot2 - marginsin(end);
            outagestmpi(c) = 1;
            outagesi(c) = outagesi(c) + 1;
        end
        if CtoNi_tot(s,c) < CtoN_tot2 - marginsin(1);
            outagestmpi_nom(c) = 1;
            outagesi_nom(c) = outagesi_nom(c) + 1;
        end
    end
end

```

```

        if CtoNi_tot(s,c) > CtoN_tot2 - marginsin(1)
            tot_duri = tot_duri + durin(1);
        elseif CtoNi_tot(s,c) > CtoN_tot2 - marginsin(2)
            tot_duri = tot_duri + durin(2);
        elseif CtoNi_tot(s,c) > CtoN_tot2 - marginsin(3)
            tot_duri = tot_duri + durin(3);
        elseif CtoNi_tot(s,c) > CtoN_tot2 - marginsin(4)
            tot_duri = tot_duri + durin(4);
        elseif CtoNi_tot(s,c) > CtoN_tot2 - marginsin(5)
            tot_duri = tot_duri + durin(5);
        elseif CtoNi_tot(s,c) > CtoN_tot2 - marginsin(6)
            tot_duri = tot_duri + durin(6);
        elseif CtoNi_tot(s,c) > CtoN_tot2 - marginsin(7)
            tot_duri = tot_duri + durin(7);
        elseif CtoNi_tot(s,c) > CtoN_tot2 - marginsin(8)
            tot_duri = tot_duri + durin(8);
        elseif CtoNi_tot(s,c) > CtoN_tot2 - marginsin(9)
            tot_duri = tot_duri + durin(9);
        end
    end
    outagestmp_nom = outagestmpo_nom | outagestmpi_nom;
    outagestmp = outagestmpo | outagestmpi;
    outages_nom = outages_nom + outagestmp_nom;
    outages = outages + outagestmp;
    if tot_duro+Nstations*guard_time > frame_duration
        overflowtmpo(s) = 1;
    end
    if tot_duri+Nstations*guard_time > frame_duration*no_trf_channels_tot
        overflowtmpi(s) = 1;
    end
    durationsi(s) = tot_duri + Nstations*guard_time;
    durationso(s) = tot_duro + Nstations*guard_time;
end
overflow = overflow + sum(overflowtmpo | overflowtmpi);
usageo = usageo + hist(durationso,xo)';
usagei = usagei + hist(durationsi,xi)';

% Gather joint statistics
sampletot = sampletot + Nstations*86400;

% Update matrix
ts40(:,1:Nstations) = ts40(:,2:Nstations+1);
ts50(:,1:Nstations) = ts50(:,2:Nstations+1);
gas40(:,1:Nstations) = gas40(:,2:Nstations+1);
gas50(:,1:Nstations) = gas50(:,2:Nstations+1);
date = daysInfo(rem(m+Nstations,len)+1).date;
dummycell = load(['Data3/S40_' date '.mat']);
eval(['ts40(:,Nstations+1) = dummycell.S40_' date '(:,2);']);
eval(['gas40(:,Nstations+1) = dummycell.S40_' date '(:,2) - dummycell.S40_' date
'(:,1);']);
dummycell = load(['Data3/S50_' date '.mat']);
eval(['ts50(:,Nstations+1) = dummycell.S50_' date '(:,2);']);
eval(['gas50(:,Nstations+1) = dummycell.S50_' date '(:,2) - dummycell.S50_' date
'(:,1);']);

currenttime(m) = toc;
totaltime = totaltime + currenttime;
remainingtime = currenttime(m)*(len-m);
[hours,mins,secs]=secs2time(remainingtime);
disp([int2str(m) ': Time remaining ' int2str(hours) 'h ' int2str(mins) 'm '
int2str(secs) 's']);
end
outagetot_nom = sum(outages_nom);
outagetot = sum(outages);
percout = 100*outagetot/sampletot;
percout_nom = 100*outagetot_nom/sampletot;
percavail = 100 - percout;
percavail_nom = 100 - percout_nom;
percoverflow = 100*overflow/(86400*len);
pdfo = hist2pdf(usageo,1e-4);
pdfi = hist2pdf(usagei,1e-4);
cdfo = 100*pdf2cdf(pdfo,1e-4);
cdfi = 100*pdf2cdf(pdfi,1e-4);
save('Final_basic_stats', 'overflow', 'outagetot_nom', 'outagetot'...
, 'currenttime', 'usageo', 'usagei', 'sampletot', 'outageso'...
, 'outageso_nom', 'outagesi', 'outagesi_nom', 'outages'...
, 'outages_nom', 'xo', 'xi', 'len');

```


Appendix S

Statistical analysis at V band with frequency diversity only

```
clear;
load('daysInfo2');
len = length(daysInfo);
x1 = 0:58;
usage = zeros(length(x1),1);
overflowtot = 0;
sampletot = 0;
totaltime = 0;
currenttime = zeros(len,1);
% design lpf & hpf
n = 40;
fs = 1;
cutoff = 20e-3;
Wn = cutoff/(fs/2);
blpf = fir1(n,Wn);
a = 1;
Nstations = 58;
CtoNo_tot = zeros(86400,Nstations);
CtoNi_tot = zeros(86400,Nstations);
outageso = zeros(Nstations,1);
outagesi = zeros(Nstations,1);
outages = zeros(Nstations,1);
outageso_nom = zeros(Nstations,1);
outagesi_nom = zeros(Nstations,1);
outages_nom = zeros(Nstations,1);
overflow = 0;
usagetmp = zeros(86400,1);
results = zeros(10,1);
ts40 = zeros(86400,Nstations+1);
ts50 = zeros(86400,Nstations+1);
gas20 = zeros(86400,Nstations+1);
gas40 = zeros(86400,Nstations+1);
gas50 = zeros(86400,Nstations+1);

% initialize matrix
for c = 1:Nstations+1
    date = daysInfo(c).date;
    dummycell = load(['Data3/S40_' date '.mat']);
    eval(['ts40(:,c) = dummycell.S40_' date '(:,2);']);
    eval(['gas40(:,c) = dummycell.S40_' date '(:,2) - dummycell.S40_' date '(:,1);']);
    dummycell = load(['Data3/S50_' date '.mat']);
    eval(['ts50(:,c) = dummycell.S50_' date '(:,2);']);
    eval(['gas50(:,c) = dummycell.S50_' date '(:,2) - dummycell.S50_' date '(:,1);']);
    [rain40,zr] = filter(blpf,a,ts40(:,c),ts40(1,c)*ones(n,1));
    rain40(1:86400-n/2) = rain40(n/2+1:86400);
    rain40(86400-n/2+1:86400) = zr(1:n/2);
    nogas40 = ts40(:,c) - gas40(:,c);
    scint40 = ts40(:,c) - rain40;
    ts14(:,c) = fscale(39.5,14,nogas40) + sscale(39.5,14,scint40,0.6,0.6,30) +
gas40(:,c)-0.6;
    ts12(:,c) = fscale(39.5,12,nogas40) + sscale(39.5,12,scint40,0.6,0.6,30) +
gas40(:,c)-0.7;
end

for m = 1:len
    tic;
    % Change single site time series to joint site time series
    timeshift20 = 14.5*60;
    timeshift40 = 14.5*60;
    timeshift50 = 14.5*60;
    for c20 = 1:86400-timeshift20
        ts12(c20,1) = min(ts12(c20,1),ts12(c20+timeshift20,1));
        ts14(c20,1) = min(ts14(c20,1),ts14(c20+timeshift20,1));
    end
    for c20 = 86400-timeshift20+1:86400
        ts12(c20,1) = min(ts12(c20,1),ts12(c20-86400+timeshift20,2));
        ts14(c20,1) = min(ts14(c20,1),ts14(c20-86400+timeshift20,2));
    end
end
```

```

end
for c40 = 1:86400-timeshift40
    ts40(c40,1) = min(ts40(c40,1),ts40(c40+timeshift40,1));
end
for c40 = 86400-timeshift40+1:86400
    ts40(c40,1) = min(ts40(c40,1),ts40(c40-86400+timeshift40,2));
end
for c50 = 1:86400-timeshift50
    ts50(c50,1) = min(ts50(c50,1),ts50(c50+timeshift50,1));
end
for c50 = 86400-timeshift50+1:86400
    ts50(c50,1) = min(ts50(c50,1),ts50(c50-86400+timeshift50,2));
end

% Load variables from link budget m-file and rename duplicates
load('OutboundTraffic_ku');
CtoN_up1 = CtoN_up;
CtoN_up_ku1 = CtoN_up_ku;
CtoN_down1 = CtoN_down;
CtoN_down_ku1 = CtoN_down_ku;
CtoN_req1 = CtoN_req;
CtoN_req_ku1 = CtoN_req_ku;
Taca1 = Taca;
Taca_ku1 = Taca_ku;
load('InboundTraffic_ku');
CtoN_up2 = CtoN_up;
CtoN_up_ku2 = CtoN_up_ku;
CtoN_down2 = CtoN_down;
CtoN_down_ku2 = CtoN_down_ku;
CtoN_req2 = CtoN_req;
CtoN_req_ku2 = CtoN_req_ku;
Taca2 = Taca;
Taca_ku2 = Taca_ku;
% Calculate Outbound Uplink C/N
CtoNo_u = CtoN_up1*ones(86400,1) - ts50(:,1);
CtoNo_u_ku = CtoN_up_ku1*ones(86400,1) - ts14(:,1);
% Calculate Outbound Downlink & Total C/N
Tsyscal = 10^(Ts_vsat/10);
Tsysca_ku1 = 10^(Ts_vsat_ku/10);
CtoNo_u_lin = 10.^(CtoNo_u/10);
CtoNo_u_lin_ku = 10.^(CtoNo_u_ku/10);
for c = 1:Nstations
    Tarain1 = 275*(1-10.^(-(gas40(:,c+1)+ts40(:,c+1))/10));
    Tarain_ku1 = 275*(1-10.^(-(gas20(:,c+1)-0.3+ts12(:,c+1))/10));
    dTsys1 = Tarain1-Taca1;
    dTsys_ku1 = Tarain_ku1-Taca_ku1;
    Tsysrain1 = Tsyscal+dTsys1;
    Tsysrain_ku1 = Tsysca_ku1+dTsys_ku1;
    dNrain1 = 10*log10(Tsysrain1/Tsyscal);
    dNrain_ku1 = 10*log10(Tsysrain_ku1/Tsysca_ku1);
    CtoNo_d = CtoN_down1 - ts40(:,c+1) - dNrain1;
    CtoNo_d_ku = CtoN_down_ku1 - ts12(:,c+1) - dNrain_ku1;
    CtoNo_lin = 1./((1./(10.^(CtoNo_d/10))) + (1./CtoNo_u_lin));
    CtoNo_tot(:,c) = 10*log10(CtoNo_lin);
    CtoNo_d_ku = 1./((1./(10.^(CtoNo_d_ku/10))) + (1./CtoNo_u_lin_ku));
    CtoNo_tot_ku(:,c) = 10*log10(CtoNo_d_ku);
end
% Calculate Inbound Downlink C/N
Tsysca2 = 10^(Ts_hub/10);
Tsysca_ku2 = 10^(Ts_hub_ku/10);
Tarain2 = 275*(1-10.^(-(gas40(:,1)+ts40(:,1))/10));
Tarain_ku2 = 275*(1-10.^(-(gas20(:,1)-0.3+ts12(:,1))/10));
dTsys2 = Tarain2-Taca2;
dTsys_ku2 = Tarain_ku2-Taca_ku2;
Tsysrain2 = Tsysca2+dTsys2;
Tsysrain_ku2 = Tsysca_ku2+dTsys_ku2;
dNrain2 = 10*log10(Tsysrain2/Tsysca2);
dNrain_ku2 = 10*log10(Tsysrain_ku2/Tsysca_ku2);
CtoNi_d = CtoN_down2 - ts40(:,1) - dNrain2;
CtoNi_d_ku = CtoN_down_ku2 - ts12(:,1) - dNrain_ku2;
% Calculate Inbound Uplink & Total C/N
CtoNi_d_lin = 10.^(CtoNi_d/10);
CtoNi_d_lin_ku = 10.^(CtoNi_d_ku/10);
for c = 1:Nstations
    CtoNi_u = CtoN_up2 - ts50(:,c+1);
    CtoNi_u_ku = CtoN_up_ku2 - ts14(:,c+1);
    CtoNi_lin = 1./((1./CtoNi_d_lin) + (1./(10.^(CtoNi_u/10))));

```

```

CtoNi_tot(:,c) = 10*log10(CtoNi_lin);
CtoNi_lin_ku = 1./((1./CtoNi_d_lin_ku) + (1./(10.^(CtoNi_u_ku/10))));
CtoNi_tot_ku(:,c) = 10*log10(CtoNi_lin_ku);
end

% Gather statistics
for s = 1:86400
    outagestmpo = zeros(Nstations,1);
    outagestmpo_nom = zeros(Nstations,1);
    outagestmpi = zeros(Nstations,1);
    outagestmpi_nom = zeros(Nstations,1);
    for c = 1:Nstations
        % Check outbound stat
        if CtoNo_tot(s,c) < CtoN_req1
            outagestmpo_nom(c) = 1;
            outageso_nom(c) = outageso_nom(c) + 1;
            if CtoNo_tot_ku(s,c) < CtoN_req_ku1
                outagestmpo(c) = 1;
                outageso(c) = outageso(c) + 1;
            end
        end
        % Check inbound stat
        if CtoNi_tot(s,c) < CtoN_req2
            outagestmpi_nom(c) = 1;
            outagesi_nom(c) = outagesi_nom(c) + 1;
            if CtoNi_tot_ku(s,c) < CtoN_req_ku2
                outagestmpi(c) = 1;
                outagesi(c) = outagesi(c) + 1;
            end
        end
        outagestmp_nom = outagestmpo_nom | outagestmpi_nom;
        outagestmp = outagestmpo | outagestmpi;
        outages_nom = outages_nom + outagestmp_nom;
        outages = outages + outagestmp;
        usagetmp(s) = sum(outagestmp_nom);
        if sum(outagestmp_nom) > no_bursts_per_channel_ku
            overflow = overflow + 1;
        end
    end
end
usage = usage + hist(usagetmp,x1)';
sampletot = sampletot + Nstations*86400;

% Update matrix
ts12(:,1:Nstations) = ts12(:,2:Nstations+1);
ts14(:,1:Nstations) = ts14(:,2:Nstations+1);
ts40(:,1:Nstations) = ts40(:,2:Nstations+1);
ts50(:,1:Nstations) = ts50(:,2:Nstations+1);
gas40(:,1:Nstations) = gas40(:,2:Nstations+1);
gas50(:,1:Nstations) = gas50(:,2:Nstations+1);
date = daysInfo(rem(m+Nstations,len)+1).date;
dummycell = load(['Data3/S40_' date '.mat']);
eval(['ts40(:,Nstations+1) = dummycell.S40_' date '(:,2);']);
eval(['gas40(:,Nstations+1) = dummycell.S40_' date '(:,2) - dummycell.S40_' date
'(:,1);']);
dummycell = load(['Data3/S50_' date '.mat']);
eval(['ts50(:,Nstations+1) = dummycell.S50_' date '(:,2);']);
eval(['gas50(:,Nstations+1) = dummycell.S50_' date '(:,2) - dummycell.S50_' date
'(:,1);']);
[rain40,zr] = filter(blpf,a,ts40(:,Nstations+1),ts40(1,Nstations+1)*ones(n,1));
rain40(1:86400-n/2) = rain40(n/2+1:86400);
rain40(86400-n/2+1:86400) = zr(1:n/2);
nogas40 = ts40(:,Nstations+1) - gas40(:,Nstations+1);
scint40 = ts40(:,Nstations+1) - rain40;
ts14(:,Nstations+1) = fscale(39.5,14,nogas40) + sscale(39.5,14,scint40,0.6,0.6,30)
+ gas40(:,Nstations+1)-0.6;
ts12(:,Nstations+1) = fscale(39.5,12,nogas40) + sscale(39.5,12,scint40,0.6,0.6,30)
+ gas40(:,Nstations+1)-0.7;

currenttime(m) = toc;
totaltime = totaltime + currenttime;
remainingtime = currenttime(m)*(len-m);
[hours,mins,secs]=secs2time(remainingtime);
disp([int2str(m) ': Time remaining ' int2str(hours) 'h ' int2str(mins) 'm '
int2str(secs) 's']);
end
outagetot_nom = sum(outages_nom);

```

```

outagetot = sum(outages);
percout = 100*outagetot/sampletot;
percout_nom = 100*outagetot_nom/sampletot;
percavail = 100 - percout;
percavail_nom = 100 - percout_nom;
percoverflow = 100*overflowtot/(86400*len);
pdf1 = hist2pdf(usage,1);
cdf1 = 100*pdf2cdf(pdf1,1);
save('Final_basic_stats_ku', 'overflow', 'outagetot_nom', 'outagetot'...
    , 'currenttime', 'usage', 'sampletot', 'outageso'...
    , 'outageso_nom', 'outagesi', 'outagesi_nom', 'outages'...
    , 'outages_nom', 'x1', 'len', 'pdf1', 'cdf1', 'percavail'...
    , 'percavail_nom', 'percoverflow', 'percout', 'percout_nom');

```

Appendix T

Statistical analysis at V band with hybrid FMT

```
clear;
load('daysInfo2');
len = length(daysInfo);
xo = 0:1e-4:50.5e-3;
xi = 0:1e-4:990.7e-3;
usageo = zeros(length(xo),1);
usagei = zeros(length(xi),1);
currenttime = zeros(len,1);
overflow = 0;
sampletot = 0;
totaltime = 0;

% design lpf & hpf
n = 40;
fs = 1;
cutoff = 20e-3;
Wn = cutoff/(fs/2);
blpf = fir1(n,Wn);
a = 1;
Nstations = 58;
CtoNo_tot = zeros(86400,Nstations);
CtoNi_tot = zeros(86400,Nstations);
outageso = zeros(Nstations,1);
outagesi = zeros(Nstations,1);
outageso_nom = zeros(Nstations,1);
outagesi_nom = zeros(Nstations,1);
outages_nom = zeros(Nstations,1);
outages = zeros(Nstations,1);
results = zeros(10,1);
ts40 = zeros(86400,Nstations+1);
ts50 = zeros(86400,Nstations+1);
gas20 = zeros(86400,Nstations+1);
gas40 = zeros(86400,Nstations+1);
gas50 = zeros(86400,Nstations+1);

% initialize matrix
for c = 1:Nstations+1
    date = daysInfo(c).date;
    dummycell = load(['Data3/S40_' date '.mat']);
    eval(['ts40(:,c) = dummycell.S40_' date '(:,2);']);
    eval(['gas40(:,c) = dummycell.S40_' date '(:,2) - dummycell.S40_' date '(:,1);']);
    dummycell = load(['Data3/S50_' date '.mat']);
    eval(['ts50(:,c) = dummycell.S50_' date '(:,2);']);
    eval(['gas50(:,c) = dummycell.S50_' date '(:,2) - dummycell.S50_' date '(:,1);']);
    [rain40,zr] = filter(blpf,a,ts40(:,c),ts40(1,c)*ones(n,1));
    rain40(1:86400-n/2) = rain40(n/2+1:86400);
    rain40(86400-n/2+1:86400) = zr(1:n/2);
    nogas40 = ts40(:,c) - gas40(:,c);
    scint40 = ts40(:,c) - rain40;
    ts14(:,c) = fscale(39.5,14,nogas40) + sscale(39.5,14,scint40,0.6,0.6,30) +
gas40(:,c)-0.6;
    ts12(:,c) = fscale(39.5,12,nogas40) + sscale(39.5,12,scint40,0.6,0.6,30) +
gas40(:,c)-0.7;
end

for m = 1:len
    tic;
    % Change single site time series to joint site time series
    timeshift20 = 14.5*60;
    timeshift40 = 14.5*60;
    timeshift50 = 14.5*60;
    for c20 = 1:86400-timeshift20
        ts12(c20,1) = min(ts12(c20,1),ts12(c20+timeshift20,1));
        ts14(c20,1) = min(ts14(c20,1),ts14(c20+timeshift20,1));
    end
    for c20 = 86400-timeshift20+1:86400
        ts12(c20,1) = min(ts12(c20,1),ts12(c20-86400+timeshift20,2));
    end
```

```

    ts14(c20,1) = min(ts14(c20,1),ts14(c20-86400+timeshift20,2));
end
for c40 = 1:86400-timeshift40
    ts40(c40,1) = min(ts40(c40,1),ts40(c40+timeshift40,1));
end
for c40 = 86400-timeshift40+1:86400
    ts40(c40,1) = min(ts40(c40,1),ts40(c40-86400+timeshift40,2));
end
for c50 = 1:86400-timeshift50
    ts50(c50,1) = min(ts50(c50,1),ts50(c50+timeshift50,1));
end
for c50 = 86400-timeshift50+1:86400
    ts50(c50,1) = min(ts50(c50,1),ts50(c50-86400+timeshift50,2));
end

load('OutboundTraffic_mixedold');
CtoN_up1 = CtoN_up;
CtoN_up_ku1 = CtoN_up_ku;
CtoN_down1 = CtoN_down;
CtoN_down_ku1 = CtoN_down_ku;
CtoN_req1 = CtoN_req;
CtoN_req_ku1 = CtoN_req_ku;
CtoN_tot1 = CtoN_tot;
Taca1 = Taca;
Taca_ku1 = Taca_ku;
load('InboundTraffic_mixedold');
CtoN_up2 = CtoN_up;
CtoN_up_ku2 = CtoN_up_ku;
CtoN_down2 = CtoN_down;
CtoN_down_ku2 = CtoN_down_ku;
CtoN_req2 = CtoN_req;
CtoN_req_ku2 = CtoN_req_ku;
CtoN_tot2 = CtoN_tot;
Taca2 = Taca;
Taca_ku2 = Taca_ku;
% Calculate Outbound Uplink C/N
CtoNo_u = CtoN_up1*ones(86400,1) - ts50(:,1);
CtoNo_u_ku = CtoN_up_ku1*ones(86400,1) - ts14(:,1);
% Calculate Outbound Downlink & Total C/N
Tsysca1 = 10^(Ts_vsats/10);
Tsysca_ku1 = 10^(Ts_vsats_ku/10);
CtoNo_u_lin = 10.^(CtoNo_u/10);
CtoNo_u_lin_ku = 10.^(CtoNo_u_ku/10);
for c = 1:Nstations
    Tarain1 = 275*(1-10.^(-(gas40(:,c+1)+ts40(:,c+1))/10));
    Tarain_ku1 = 275*(1-10.^(-(gas20(:,c+1)-0.3+ts12(:,c+1))/10));
    dTsys1 = Tarain1-Taca1;
    dTsys_ku1 = Tarain_ku1-Taca_ku1;
    Tsyrain1 = Tsysca1+dTsys1;
    Tsyrain_ku1 = Tsysca_ku1+dTsys_ku1;
    dNrain1 = 10*log10(Tsyrain1/Tsysca1);
    dNrain_ku1 = 10*log10(Tsyrain_ku1/Tsysca_ku1);
    CtoNo_d = CtoN_down1 - ts40(:,c+1) - dNrain1;
    CtoNo_d_ku = CtoN_down_ku1 - ts12(:,c+1) - dNrain_ku1;
    CtoNo_lin = 1./((1./(10.^(CtoNo_d/10))) + (1./CtoNo_u_lin));
    CtoNo_tot(:,c) = 10*log10(CtoNo_lin);
    CtoNo_lin_ku = 1./((1./(10.^(CtoNo_d_ku/10))) + (1./CtoNo_u_lin_ku));
    CtoNo_tot_ku(:,c) = 10*log10(CtoNo_lin_ku);
end
% Calculate Inbound Downlink C/N
Tsysca2 = 10^(Ts_hub/10);
Tsysca_ku2 = 10^(Ts_hub_ku/10);
Tarain2 = 275*(1-10.^(-(gas40(:,1)+ts40(:,1))/10));
Tarain_ku2 = 275*(1-10.^(-(gas20(:,1)-0.3+ts12(:,1))/10));
dTsys2 = Tarain2-Taca2;
dTsys_ku2 = Tarain_ku2-Taca_ku2;
Tsyrain2 = Tsysca2+dTsys2;
Tsyrain_ku2 = Tsysca_ku2+dTsys_ku2;
dNrain2 = 10*log10(Tsyrain2/Tsysca2);
dNrain_ku2 = 10*log10(Tsyrain_ku2/Tsysca_ku2);
CtoNi_d = CtoN_down2 - ts40(:,1) - dNrain2;
CtoNi_d_ku = CtoN_down_ku2 - ts12(:,1) - dNrain_ku2;
% Calculate Inbound Uplink & Total C/N
CtoNi_d_lin = 10.^(CtoNi_d/10);
CtoNi_d_lin_ku = 10.^(CtoNi_d_ku/10);
for c = 1:Nstations
    CtoNi_u = CtoN_up2 - ts50(:,c+1);

```

```

CtoNi_u_ku = CtoN_up_ku2 - ts14(:,c+1);
CtoNi_lin = 1./((1./CtoNi_d_lin) + (1./(10.^(CtoNi_u/10))));
CtoNi_tot(:,c) = 10*log10(CtoNi_lin);
CtoNi_lin_ku = 1./((1./CtoNi_d_lin_ku) + (1./(10.^(CtoNi_u_ku/10))));
CtoNi_tot_ku(:,c) = 10*log10(CtoNi_lin_ku);
end

% Gather statistics
overflowtmpo = zeros(86400,1);
overflowtmpi = zeros(86400,1);
durationso = zeros(86400,1);
durationsi = zeros(86400,1);
for s = 1:86400
    outagestmpo = zeros(Nstations,1);
    outagestmpo_nom = zeros(Nstations,1);
    outagestmpi = zeros(Nstations,1);
    outagestmpi_nom = zeros(Nstations,1);
    tot_duro = 0;
    tot_duri = 0;
    % Determine which stations are using ku band
    ind_ku = zeros(no_bursts_per_channel_ku,1);
    val_ku = 10000*ones(no_bursts_per_channel_ku,1);
    for c = 1:Nstations
        mc = 1;
        notfilled = 1;
        if CtoNi_tot_ku(s,c) > CtoN_req_ku2
            while notfilled && mc <= 4
                if CtoNi_tot_ku(s,c) < val_ku(mc)
                    val_ku(mc+1:no_bursts_per_channel_ku)=val_ku(mc:no_bursts_per_channel_ku-1);
                    val_ku(mc) = CtoNi_tot_ku(s,c);

                    ind_ku(mc+1:no_bursts_per_channel_ku)=ind_ku(mc:no_bursts_per_channel_ku-1);
                    ind_ku(mc) = c;
                    notfilled = 0;
                end
                mc = mc + 1;
            end
        end
    end
    for c = 1:Nstations
        % Check outbound stat
        if CtoNo_tot_ku(s,c) < CtoN_req_ku1
            outagestmpo(c) = 1;
            outageso(c) = outageso(c) + 1;
        end
        if CtoNo_tot(s,c) < CtoN_tot1 - marginsout(1);
            outagestmpo_nom(c) = 1;
            outageso_nom(c) = outageso_nom(c) + 1;
        end
        if ~any(ind_ku == c)
            if CtoNo_tot(s,c) > CtoN_tot1 - marginsout(1)
                tot_duro = tot_duro + durout(1);
            elseif CtoNo_tot(s,c) > CtoN_tot1 - marginsout(2)
                tot_duro = tot_duro + durout(2);
            elseif CtoNo_tot(s,c) > CtoN_tot1 - marginsout(3)
                tot_duro = tot_duro + durout(3);
            elseif CtoNo_tot(s,c) > CtoN_tot1 - marginsout(4)
                tot_duro = tot_duro + durout(4);
            elseif CtoNo_tot(s,c) > CtoN_tot1 - marginsout(5)
                tot_duro = tot_duro + durout(5);
            elseif CtoNo_tot(s,c) > CtoN_tot1 - marginsout(6)
                tot_duro = tot_duro + durout(6);
            elseif CtoNo_tot(s,c) > CtoN_tot1 - marginsout(7)
                tot_duro = tot_duro + durout(7);
            elseif CtoNo_tot(s,c) > CtoN_tot1 - marginsout(8)
                tot_duro = tot_duro + durout(8);
            elseif CtoNo_tot(s,c) > CtoN_tot1 - marginsout(9)
                tot_duro = tot_duro + durout(9);
            end
        end
        if CtoNi_tot_ku(s,c) < CtoN_req_ku2
            outagestmpi(c) = 1;
            outagesi(c) = outagesi(c) + 1;
        end
        if CtoNi_tot(s,c) < CtoN_tot2 - marginsin(1);
            outagestmpi_nom(c) = 1;

```

```

        outagesi_nom(c) = outagesi_nom(c) + 1;
    end
    if ~any(ind_ku == c)
        if CtoNi_tot(s,c) > CtoN_tot2 - marginsin(1)
            tot_duri = tot_duri + durin(1);
        elseif CtoNi_tot(s,c) > CtoN_tot2 - marginsin(2)
            tot_duri = tot_duri + durin(2);
        elseif CtoNi_tot(s,c) > CtoN_tot2 - marginsin(3)
            tot_duri = tot_duri + durin(3);
        elseif CtoNi_tot(s,c) > CtoN_tot2 - marginsin(4)
            tot_duri = tot_duri + durin(4);
        elseif CtoNi_tot(s,c) > CtoN_tot2 - marginsin(5)
            tot_duri = tot_duri + durin(5);
        elseif CtoNi_tot(s,c) > CtoN_tot2 - marginsin(6)
            tot_duri = tot_duri + durin(6);
        elseif CtoNi_tot(s,c) > CtoN_tot2 - marginsin(7)
            tot_duri = tot_duri + durin(7);
        elseif CtoNi_tot(s,c) > CtoN_tot2 - marginsin(8)
            tot_duri = tot_duri + durin(8);
        elseif CtoNi_tot(s,c) > CtoN_tot2 - marginsin(9)
            tot_duri = tot_duri + durin(9);
        end
    end
    end
    outagetsmp_nom = outagetsmpo_nom | outagetsmpi_nom;
    outagetsmp = outagetsmpo | outagetsmpi;
    outages_nom = outages_nom + outagetsmp_nom;
    outages = outages + outagetsmp;
    if tot_duro+Nstations*guard_time > frame_duration
        overflowtmpo(s) = 1;
    end
    if tot_duri+Nstations*guard_time > frame_duration*no_trf_channels_tot
        overflowtmpi(s) = 1;
    end
    durationsi(s) = tot_duri + (Nstations-4*no_ku_channels)*guard_time;
    durationso(s) = tot_duro + (Nstations-4*no_ku_channels)*guard_time;
end
overflow = overflow + sum(overflowtmpo | overflowtmpi);
usageo = usageo + hist(durationsso,xo)';
usagei = usagei + hist(durationsi,xi)';

% Gather joint statistics
sampletot = sampletot + Nstations*86400;

% Update matrix
ts12(:,1:Nstations) = ts12(:,2:Nstations+1);
ts14(:,1:Nstations) = ts14(:,2:Nstations+1);
ts40(:,1:Nstations) = ts40(:,2:Nstations+1);
ts50(:,1:Nstations) = ts50(:,2:Nstations+1);
gas40(:,1:Nstations) = gas40(:,2:Nstations+1);
gas50(:,1:Nstations) = gas50(:,2:Nstations+1);
date = daysInfo(rem(m+Nstations,len)+1).date;
dummycell = load(['Data3/S40_' date '.mat']);
eval(['ts40(:,Nstations+1) = dummycell.S40_' date '(:,2);']);
eval(['gas40(:,Nstations+1) = dummycell.S40_' date '(:,2) - dummycell.S40_' date
'(:,1);']);
dummycell = load(['Data3/S50_' date '.mat']);
eval(['ts50(:,Nstations+1) = dummycell.S50_' date '(:,2);']);
eval(['gas50(:,Nstations+1) = dummycell.S50_' date '(:,2) - dummycell.S50_' date
'(:,1);']);
[rain40,zr] = filter(blpf,a,ts40(:,Nstations+1),ts40(1,Nstations+1)*ones(n,1));
rain40(1:86400-n/2) = rain40(n/2+1:86400);
rain40(86400-n/2+1:86400) = zr(1:n/2);
nogas40 = ts40(:,Nstations+1) - gas40(:,Nstations+1);
scint40 = ts40(:,Nstations+1) - rain40;
ts14(:,Nstations+1) = fscale(39.5,14,nogas40) + sscale(39.5,14,scint40,0.6,0.6,30)
+ gas40(:,Nstations+1)-0.6;
ts12(:,Nstations+1) = fscale(39.5,12,nogas40) + sscale(39.5,12,scint40,0.6,0.6,30)
+ gas40(:,Nstations+1)-0.7;

currenttime(m) = toc;
totaltime = totaltime + currenttime;
remainingtime = currenttime(m)*(len-m);
[hours,mins,secs]=secs2time(remainingtime);
disp([int2str(m) ' : Time remaining ' int2str(hours) 'h ' int2str(mins) 'm '
int2str(secs) 's']);
end

```



```

outagetot_nom = sum(outages_nom);
outagetot = sum(outages);
percout = 100*outagetot/sampletot;
percout_nom = 100*outagetot_nom/sampletot;
percavail = 100 - percout;
percavail_nom = 100 - percout_nom;
percoverflow = 100*overflow/(86400*len);
pdfo = hist2pdf(usageo,1e-4);
pdfi = hist2pdf(usagi,1e-4);
cdfo = 100*pdf2cdf(pdfo,1e-4);
cdfi = 100*pdf2cdf(pdfi,1e-4);
save('Final_basic_stats_mixed', 'overflow', 'outagetot_nom', 'outagetot'...
    , 'currenttime', 'usageo', 'usagi', 'sampletot', 'outageso'...
    , 'outageso_nom', 'outagesi', 'outagesi_nom', 'outages'...
    , 'outages_nom', 'xo', 'xi', 'len');

```

Appendix U1

Final statistical analysis with 4 ku band stations

```
clear;
load('daysInfo2');
load('inter_session_indices_620');
load('SystemDefinitions');
load('Fwd_ctrl_info','Rs_fwd_ctrl','B_fwd_ctrl');
load('Rtn_ctrl_info','Rs_rtn_ctrl','B_rtn_ctrl');
load('Fwd_traf_info','Rs_fwd_traf','Rs_fwd_traf_ku','B','B_ku');
load('Rtn_traf_info','Rs_rtn_traf','Rs_rtn_traf_ku','B_per_channel','B_per_channel_ku');
load('OutboundTraffic_mixed','CtoN_req','CtoN_req_ku','CtoN_tot','CtoN_tot_ku','durout',
    'margin_sout');
CNR_req_fwd = CtoN_req;
CNR_req_fwd_ku = CtoN_req_ku;
CNR_ca_fwd = CtoN_tot;
CNR_ca_fwd_ku = CtoN_tot_ku;
duration_fwd = durout;
margin_fwd = margin_sout;
load('InboundTraffic_mixed','CtoN_req','CtoN_req_ku','CtoN_tot','CtoN_tot_ku','durin',
    'margin_sin');
CNR_req_rtn = CtoN_req;
CNR_req_rtn_ku = CtoN_req_ku;
CNR_ca_rtn = CtoN_tot;
CNR_ca_rtn_ku = CtoN_tot_ku;
duration_rtn = durin;
margin_rtn = margin_sin;
clear CtoN_req CtoN_req_ku CtoN_tot CtoN_tot_ku durin durout margin_sin margin_sout;
load('Final_fixed_margin_CNRs');

% System info
Nstations = find(activity_cdf==0,1)-1;
Phub_traf_V = 5;
Phub_traf_ku = 199;
Phub_ctrl = 1;
Pvsat_traf_V = 0.8;
Pvsat_traf_ku = 2*0.8;
Pvsat_ctrl = 2*0.2;
TransponderBW = 36e6;
Psat_fwd_traf = Psat*B/TransponderBW;
Psat_rtn_traf = Psat*B_per_channel/TransponderBW;
Psat_fwd_traf_ku = Psat*B_ku/TransponderBW;
Psat_rtn_traf_ku = Psat*B_per_channel_ku/TransponderBW;
Psat_fwd_ctrl = Psat*B_fwd_ctrl/TransponderBW;
Psat_rtn_ctrl = Psat*B_rtn_ctrl/TransponderBW;
Psat_fwd_traf_fixed = Psat*B_fwd_fixed/TransponderBW;
Psat_rtn_traf_fixed = Psat*B_per_channel_fixed/TransponderBW;
Tsysca_sat_V = 900;
Tsysca_sat_ku = 500;
Tsysca_ground_V = 350;
Tsysca_ground_ku = 150;
Leoc_hub = 1;
Leoc_vsat = 3;
Lmisc_hub = 1;
Lmisc_vsat = 1;
FMT_CNR_reqs = [8.5 7.9 7.2 6.7 5.8 4.9 4.2 3.7 2.8];
FMT_code_rates = {'6/7','4/5','3/4','2/3','1/2','4/5','3/4','2/3','1/2'};
FMT_bits_per_symbol = [2 2 2 2 1 1 1 1];
no_bursts_per_channel_ku = 1;
Gdif = Grx_hub_ku - Grx_vsat_ku;
scint_margin = 2;
scint_margin_ku = 1;
gas14 = 0.7;
gas12 = 0.5;

% design lpf & hpf
n = 40;
fs = 1;
cutoff = 20e-3;
```

```

Wn = cutoff/(fs/2);
blpf = fir1(n,Wn);
a = 1;

% Initialize working memory
ts40 = zeros(86400,Nstations+1);
ts50 = zeros(86400,Nstations+1);
gas40 = zeros(86400,Nstations+1);
gas50 = zeros(86400,Nstations+1);
ts14 = zeros(86400,Nstations+1);
ts12 = zeros(86400,Nstations+1);
CNR_fwd_down_V = zeros(86400,Nstations);
CNR_rtn_up_V = zeros(86400,Nstations);
CNR_fwd_down_ku = zeros(86400,Nstations);
CNR_rtn_up_ku = zeros(86400,Nstations);
CNR_fwd_down_fixed = zeros(86400,Nstations);
CNR_rtn_up_fixed = zeros(86400,Nstations);
CNR_fwd_V = zeros(86400,Nstations);
CNR_rtn_V = zeros(86400,Nstations);
CNR_fwd_ku = zeros(86400,Nstations);
CNR_rtn_ku = zeros(86400,Nstations);
CNR_fwd_fixed = zeros(86400,Nstations);
CNR_rtn_fixed = zeros(86400,Nstations);
CNR_fwd_pred_V = zeros(86400,Nstations);
CNR_fwd_pred_ku = zeros(86400,Nstations);
active_station_info = zeros(Nstations,1);
errors_rcst_ind = zeros(8,1);
call_count = ones(Nstations_tot,1);
[len,dum] = size(inter_session_start);
event_count = 0;
channel_type = zeros(86400,Nstations);
csec = 0;
xh = 0:0.001:1;
outages_fading = 0;
outages_fading_fixed = 0;
outages_fading_fixed_nomargin = 0;
outages_noresource = 0;
no_active_station_seconds = 0;
event_no = 0;
station_outage_count = zeros(Nstations_tot,1);
outages_fwd = zeros(Nstations_tot,1);
outages_fwd_fixed = zeros(Nstations_tot,1);
burst_duration_fwd = zeros(1,1001);
burst_duration_fwd_req = zeros(1,1001);
outages_fixed = zeros(Nstations_tot,1);
outages_fixed_nomargin = zeros(Nstations_tot,1);
traf_fmt_ratio = zeros(1,101);

% Create lookup table for dNrain, indexed by attenuation
dNrain_ku = zeros(101,1);
CNR_reduction_ku = zeros(101,1);
CNR_reduction_hub_ku = zeros(101,1);
Taca_ku = 275*(1-10^(-(gas12/10)));
for att = 0:100
    Tarain_ku = 275*(1-10^(-(gas12+att)/10));
    dTsys_ku = Tarain_ku - Taca_ku;
    Tsysrain_ku = Tsysca_ground_ku + dTsys_ku;
    dNrain_ku(att+1) = 10*log10(Tsysrain_ku/Tsysca_ground_ku);
    CNR_reduction_ku(att+1) = att + dNrain_ku(att+1);
    CNR_reduction_hub_ku(att+1) = fscale(12,14,att) + att + dNrain_ku(att+1);
end
Taca_V = 275*(1-10^(-(1/10)));

% Simulation info
no_days = length(daysInfo);

for c = 1:Nstations+1
    date = daysInfo(c).date;
    dummycell = load(['Data3/S40_' date '.mat']);
    eval(['ts40(:,c) = dummycell.S40_' date '(:,2);']);
    eval(['gas40(:,c) = dummycell.S40_' date '(:,2) - dummycell.S40_' date '(:,1);']);
    dummycell = load(['Data3/S50_' date '.mat']);
    eval(['ts50(:,c) = dummycell.S50_' date '(:,2);']);
    eval(['gas50(:,c) = dummycell.S50_' date '(:,2) - dummycell.S50_' date '(:,1);']);
    [rain40,zr] = filter(blpf,a,ts40(:,c),ts40(1,c)*ones(n,1));
    rain40(1:86400-n/2) = rain40(n/2+1:86400);
    rain40(1:n/2) = ts40(1:n/2,c);

```

```

rain40(86400-n/2+1:86400) = ts40(86400-n/2+1:86400,c);
nogas40 = rain40 - gas40(:,c);
scint40 = ts40(:,c) - rain40;
ts14(:,c) = fscale(39.5,14,nogas40) + sscale(39.5,14,scint40,0.6,0.6,30) + gas14;
ts12(:,c) = fscale(39.5,12,nogas40) + sscale(39.5,12,scint40,0.6,0.6,30) + gas12;
end

for day = 1:no_days % Loop over all days in daysinfo
    event_time = zeros(Nstations,1);
    event_flag = zeros(Nstations,1);
    min_CNR = 20*ones(Nstations,1);
    tic;
    disp(['Doing day ' int2str(day)]);
    % Change single site time series to joint site time series
    timeshift = 14.5*60;
    for c = 1:86400-timeshift
        ts12(c,1) = min(ts12(c,1),ts12(c+timeshift,1));
        ts14(c,1) = min(ts14(c,1),ts14(c+timeshift,1));
        ts40(c,1) = min(ts40(c,1),ts40(c+timeshift,1));
        ts50(c,1) = min(ts50(c,1),ts50(c+timeshift,1));
    end
    for c = 86400-timeshift+1:86400
        ts12(c,1) = min(ts12(c,1),ts12(c-86400+timeshift,2));
        ts14(c,1) = min(ts14(c,1),ts14(c-86400+timeshift,2));
        ts40(c,1) = min(ts40(c,1),ts40(c-86400+timeshift,2));
        ts50(c,1) = min(ts50(c,1),ts50(c-86400+timeshift,2));
    end
    disp(['Site Diversity: ' num2str(toc)]);
    tic;
    % Calculate hub side CNRs
    CNR_fwd_up_V =
att2cnr(ts50(:,1),gas50(:,1),Gtx_hub_V+10*log10(Phub_traf_V),Grx_satellite_V,Tsysca_sat_V,ulink_freq,Rs_fwd_traf,Leoc_hub,Lmisc_hub,0);
    CNR_rtn_down_V =
att2cnr(ts40(:,1),gas40(:,1),Gtx_satellite_V+10*log10(Psat_rtn_traf),Grx_hub_V,Tsysca_ground_V,dlink_freq,Rs_rtn_traf,Leoc_hub,Lmisc_hub,1);
    CNR_fwd_up_ku =
att2cnr(ts14(:,1),gas14*ones(86400,1),Gtx_hub_ku+10*log10(Phub_traf_ku),Grx_satellite_ku,Tsysca_sat_ku,ulink_freq_ku,Rs_fwd_traf_ku,Leoc_hub,Lmisc_hub,0);
    CNR_rtn_down_ku =
att2cnr(ts12(:,1),gas12*ones(86400,1),Gtx_satellite_ku+10*log10(Psat_rtn_traf_ku),Grx_hub_ku,Tsysca_ground_ku,dlink_freq_ku,Rs_rtn_traf_ku,Leoc_hub,Lmisc_hub,1);
    CNR_fwd_up_fixed =
att2cnr(ts50(:,1),gas50(:,1),Gtx_hub_V+10*log10(Phub_traf_V),Grx_satellite_V,Tsysca_sat_V,ulink_freq,Rs_fwd_fixed,Leoc_hub,Lmisc_hub,0);
    CNR_rtn_down_fixed =
att2cnr(ts40(:,1),gas40(:,1),Gtx_satellite_V+10*log10(Psat_rtn_traf),Grx_hub_V,Tsysca_ground_V,dlink_freq,Rs_rtn_fixed,Leoc_hub,Lmisc_hub,1);
    CNR_fwd_up_ku0 =
att2cnr(gas14,gas14,Gtx_hub_ku+10*log10(Phub_traf_ku),Grx_satellite_ku,Tsysca_sat_ku,ulink_freq_ku,Rs_fwd_traf_ku,Leoc_hub,Lmisc_hub,0);
    CNR_fwd_up_V0 =
att2cnr(gas50(:,1),gas50(:,1),Gtx_hub_V+10*log10(Phub_traf_V),Grx_satellite_V,Tsysca_sat_V,ulink_freq,Rs_fwd_traf,Leoc_hub,Lmisc_hub,0);

    % Calculate forward downlink for the hub.
    CNR_fwd_down_hub_ku =
att2cnr(ts12(:,1),gas12*ones(86400,1),Gtx_satellite_ku+10*log10(Psat_fwd_traf_ku),Grx_hub_ku,Tsysca_ground_ku,dlink_freq_ku,Rs_fwd_traf_ku,Leoc_hub,Lmisc_hub,1);
    CNR_fwd_down_hub_ku0 =
att2cnr(gas12,gas12,Gtx_satellite_ku+10*log10(Psat_fwd_traf_ku),Grx_hub_ku,Tsysca_ground_ku,dlink_freq_ku,Rs_fwd_traf_ku,Leoc_hub,Lmisc_hub,1);
    CNR_fwd_hub_0 = 10*log10(1./((1./(10.^(CNR_fwd_up_ku0/10))) +
(1./10.^(CNR_fwd_down_hub_ku0/10))));
    CNR_fwd_hub_ku = 10*log10(1./((1./(10.^(CNR_fwd_up_ku/10))) +
(1./10.^(CNR_fwd_down_hub_ku/10))));
    CNR_fwd_pred_hub_ku = rtadalinemex(CNR_fwd_hub_ku,3,1,1e-6,[0.3 0.3
0.4],CNR_fwd_hub_ku(1));

    AttNoise_ku = CNR_fwd_hub_0 - CNR_fwd_pred_hub_ku; %CNR_ca_fwd_ku
    Att_ku = interp1(CNR_reduction_hub_ku,(0:100)',AttNoise_ku,'linear','extrap');
    Att_V = fscale(12,50,Att_ku);

    % Adjust overall CNR based on power control
    CNR_fwd_up_V_pctrl = CNR_fwd_up_V;
    for c = 1:86399

```

```

        if Att_V(c) > 6
            CNR_fwd_up_V_pctrl(c+1) = CNR_fwd_up_V(c+1) + 6;
        elseif Att_V(c) > 3
            CNR_fwd_up_V_pctrl(c+1) = CNR_fwd_up_V(c+1) + 3;
        end
    end
    disp(['Power Control: ' num2str(toc)]);
    tic;

    % Do vsat side CNR calculations & total CNR calculations with power control
    for c = 1:Nstations
        CNR_fwd_down_V(:,c) =
att2cnr(ts40(:,c+1),gas40(:,c+1),Gtx_satellite_V+10*log10(Psat_fwd_traf),Grx_vsat_V,Ts
ysca_ground_V,dlink_freq,Rs_fwd_traf,Leoc_vsat,Lmisc_vsat,1);
        CNR_rtn_up_V(:,c) =
att2cnr(ts50(:,c+1),gas50(:,c+1),Gtx_vsat_V+10*log10(Pvsat_traf_V),Grx_satellite_V,Tsy
sca_sat_V,ulink_freq,Rs_rtn_traf,Leoc_vsat,Lmisc_vsat,0);
        CNR_fwd_down_ku(:,c) =
att2cnr(ts12(:,c+1),gas12*ones(86400,1),Gtx_satellite_ku+10*log10(Psat_fwd_traf_ku),Gr
x_vsat_ku,Tsysca_ground_ku,dlink_freq_ku,Rs_fwd_traf_ku,Leoc_vsat,Lmisc_vsat,1);
        CNR_rtn_up_ku(:,c) =
att2cnr(ts14(:,c+1),gas14*ones(86400,1),Gtx_vsat_ku+10*log10(Pvsat_traf_ku),Grx_satell
ite_ku,Tsysca_sat_ku,ulink_freq_ku,Rs_rtn_traf_ku,Leoc_vsat,Lmisc_vsat,0);
        CNR_fwd_down_fixed(:,c) =
att2cnr(ts40(:,c+1),gas40(:,c+1),Gtx_satellite_V+10*log10(Psat_fwd_traf),Grx_vsat_V,Ts
ysca_ground_V,dlink_freq,Rs_fwd_fixed,Leoc_vsat,Lmisc_vsat,1);
        CNR_rtn_up_fixed(:,c) =
att2cnr(ts50(:,c+1),gas50(:,c+1),Gtx_vsat_V+10*log10(Pvsat_traf_V),Grx_satellite_V,Tsy
sca_sat_V,ulink_freq,Rs_rtn_fixed,Leoc_vsat,Lmisc_vsat,0);
        CNR_fwd_down_ku0(:,c) =
att2cnr(gas12,gas12,Gtx_satellite_ku+10*log10(Psat_fwd_traf_ku),Grx_vsat_ku,Tsysca_gro
und_ku,dlink_freq_ku,Rs_fwd_traf_ku,Leoc_vsat,Lmisc_vsat,1);
        CNR_fwd_down_V0(:,c) =
att2cnr(gas40(:,c+1),gas40(:,c+1),Gtx_satellite_V+10*log10(Psat_fwd_traf),Grx_vsat_V,T
sysca_ground_V,dlink_freq,Rs_fwd_traf,Leoc_vsat,Lmisc_vsat,1);

        CNR_fwd_V(:,c) = 10*log10(1./((1./(10.^(CNR_fwd_up_V_pctrl/10))) +
(1./10.^(CNR_fwd_down_V(:,c)/10))));
        CNR_rtn_V(:,c) = 10*log10(1./((1./(10.^(CNR_rtn_up_V(:,c)/10))) +
(1./10.^(CNR_rtn_down_V/10))));
        CNR_fwd_ku(:,c) = 10*log10(1./((1./(10.^(CNR_fwd_up_ku/10))) +
(1./10.^(CNR_fwd_down_ku(:,c)/10))));
        CNR_rtn_ku(:,c) = 10*log10(1./((1./(10.^(CNR_rtn_up_ku(:,c)/10))) +
(1./10.^(CNR_rtn_down_ku/10))));
        CNR_tmp = 10*log10(1./((1./(10.^(CNR_fwd_up_fixed/10))) +
(1./10.^(CNR_fwd_down_fixed(:,c)/10))));
        CNR_fwd_fixed(:,c) = filter(blpf,a,CNR_tmp);
        CNR_fwd_fixed(1:86400-n/2) = CNR_fwd_fixed(n/2+1:86400);
        CNR_fwd_fixed(86400-n/2+1:86400) = CNR_tmp(86400-n/2+1:86400);
        CNR_tmp = 10*log10(1./((1./(10.^(CNR_rtn_up_fixed(:,c)/10))) +
(1./10.^(CNR_rtn_down_fixed/10))));
        CNR_rtn_fixed(:,c) = filter(blpf,a,CNR_tmp);
        CNR_rtn_fixed(1:86400-n/2) = CNR_rtn_fixed(n/2+1:86400);
        CNR_rtn_fixed(86400-n/2+1:86400) = CNR_tmp(86400-n/2+1:86400);
        CNR_fwd_pred_ku(:,c) = rtadalinemex(CNR_fwd_ku(:,c),3,4,1e-6,[0.3 0.3
0.4],CNR_fwd_ku(1,c));
        CNR_fwd_pred_V(:,c) = rtadalinemex(CNR_fwd_V(:,c),3,4,1e-6,[0.3 0.3
0.4],CNR_fwd_V(1,c));
    end
    disp(['VSAT CNRs & Predictions: ' num2str(toc)]);

    tic;
    for dsec = 1:86400
        csec = csec+1;
        if dsec==1
            esec = 1;
        else
            esec = dsec - 1;
        end
        if dsec < 4
            fsec = 4;
        else
            fsec = dsec;
        end
        % Assign active station with CNR from one of the active channels
        for c = 1:Nstations_tot
            if csec == inter_session_start(call_count(c),c)

```

```

        active_station_info(find(active_station_info==0,1)) = c;
    end
    if csec == inter_session_end(call_count(c),c)
        active_station_info(find(active_station_info==c,1)) = 0;
        if call_count(c) < len
            call_count(c) = call_count(c) + 1;
        end
    end
end
end
active_index = find(active_station_info~=0);
active_sum = length(active_index);
if active_sum
    no_active_station_seconds = no_active_station_seconds + active_sum;
    % Determine which stations are using ku band
    ind_ku = linspace(1,no_bursts_per_channel_ku,no_bursts_per_channel_ku);
    val_ku = 10000*ones(no_bursts_per_channel_ku,1);
    outage_vector = zeros(active_sum,1);
    for c = 1:active_sum
        mc = 1;
        notfilled = 1;
        if any(CNR_fwd_pred_ku(fsec-3:fsec,active_index(c)) > CNR_req_fwd_ku)
            while notfilled && mc <= no_bursts_per_channel_ku
                if mean(CNR_fwd_pred_V(fsec-3:fsec,active_index(c))) <
val_ku(mc)
                val_ku(mc+1:no_bursts_per_channel_ku)=val_ku(mc:no_bursts_per_channel_ku-1);
                val_ku(mc) = mean(CNR_fwd_pred_ku(fsec-
3:fsec,active_index(c)));
            end_ku(mc+1:no_bursts_per_channel_ku)=ind_ku(mc:no_bursts_per_channel_ku-1);
                ind_ku(mc) = c;
                notfilled = 0;
            end
            mc = mc + 1;
        end
    end
    end
    outagestmp_fading = zeros(Nstations,1);
    outagestmp_fading_fixed = zeros(Nstations,1);
    outagestmp_fading_fixed_nomargin = zeros(Nstations,1);
    outagestmp_rtn_fixed = zeros(Nstations,1);
    burst_duration_fwd_temp = zeros(Nstations,1);
    burst_duration_rtn_temp = zeros(Nstations,1);
    for c = 1:active_sum
        if ~event_flag(active_index(c)) &&
CNR_fwd_pred_ku(dsec,active_index(c)) < CNR_req_fwd_ku
            event_flag(active_index(c)) = 1;
        end
        if event_flag(active_index(c))
            outagestmp_fading(active_index(c)) = 1;
            outages_fwd(active_station_info(active_index(c))) =
outages_fwd(active_station_info(active_index(c))) + 1;
            if all(CNR_fwd_pred_ku(fsec-3:fsec,active_index(c)) >
CNR_req_fwd_ku)
                event_flag(active_index(c)) = 0;
            end
        end
    end

    if CNR_fwd_fixed(dsec,active_index(c)) < CNR_req_fwd_fixed;
        outagestmp_fading_fixed_nomargin(active_index(c)) = 1;
        outages_fixed_nomargin(active_station_info(active_index(c))) =
outages_fixed_nomargin(active_station_info(active_index(c))) + 1;
    end
    % Find burst duration for each station using V-band
    if ~any(ind_ku == c)
        if all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(1))
            burst_duration_fwd_temp(active_index(c)) =
duration_fwd(1);
        elseif all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(2))
            burst_duration_fwd_temp(active_index(c)) =
duration_fwd(2);
        elseif all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(3))

```

```

        burst_duration_fwd_temp(active_index(c)) =
duration_fwd(3);
        elseif all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(4))
        burst_duration_fwd_temp(active_index(c)) =
duration_fwd(4);
        elseif all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(5))
        burst_duration_fwd_temp(active_index(c)) =
duration_fwd(5);
        elseif all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(6))
        burst_duration_fwd_temp(active_index(c)) =
duration_fwd(6);
        elseif all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(7))
        burst_duration_fwd_temp(active_index(c)) =
duration_fwd(7);
        elseif all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(8))
        burst_duration_fwd_temp(active_index(c)) =
duration_fwd(8);
        elseif all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(9))
        burst_duration_fwd_temp(active_index(c)) =
duration_fwd(9);
    end
    end
    % Check return channel
    % Find absolute outage at ku-band
end
outages_fading = outages_fading + sum(outagestmp_fading);
outages_fading_fixed = outages_fading_fixed +
sum(outagestmp_fading_fixed);
outages_fading_fixed_nomargin = outages_fading_fixed_nomargin +
sum(outagestmp_fading_fixed_nomargin);

burst_duration_fwd_all_stations = sum(burst_duration_fwd_temp);
active_sum_modified = active_sum;
burst_duration_fwd_req = burst_duration_fwd_req +
hist(sum(burst_duration_fwd_temp),xh);
while burst_duration_fwd_all_stations > 20e-3
    [max_dur,max_ind] = max(burst_duration_fwd_temp);
    burst_duration_fwd_temp(max_ind) = 0;
    burst_duration_fwd_all_stations = burst_duration_fwd_all_stations -
max_dur;

    outages_noresource = outages_noresource + 1;
    active_sum_modified = active_sum_modified - 1;
end
burst_duration_fwd = burst_duration_fwd +
hist(sum(burst_duration_fwd_temp),xh);
dur_ratio = (burst_duration_fwd_all_stations -
active_sum_modified*duration_fwd(1))/burst_duration_fwd_all_stations;
traf_fmt_ratio = traf_fmt_ratio + hist(dur_ratio,0:0.001:0.1);
end
end
disp(['Stats: ' num2str(toc)]);
tic;
% Update matrix
ts12(:,1:Nstations) = ts12(:,2:Nstations+1);
ts14(:,1:Nstations) = ts14(:,2:Nstations+1);
ts40(:,1:Nstations) = ts40(:,2:Nstations+1);
ts50(:,1:Nstations) = ts50(:,2:Nstations+1);
gas40(:,1:Nstations) = gas40(:,2:Nstations+1);
gas50(:,1:Nstations) = gas50(:,2:Nstations+1);
date = daysInfo(rem(day+Nstations,no_days)+1).date;
dummycell = load(['Data3/S40_' date '.mat']);
eval(['ts40(:,Nstations+1) = dummycell.S40_' date '(:,2);']);
eval(['gas40(:,Nstations+1) = dummycell.S40_' date '(:,2) - dummycell.S40_' date
'(:,1);']);
dummycell = load(['Data3/S50_' date '.mat']);
eval(['ts50(:,Nstations+1) = dummycell.S50_' date '(:,2);']);
eval(['gas50(:,Nstations+1) = dummycell.S50_' date '(:,2) - dummycell.S50_' date
'(:,1);']);
[rain40,zr] = filter(blpf,a,ts40(:,Nstations+1),ts40(1,Nstations+1)*ones(n,1));
rain40(1:86400-n/2) = rain40(n/2+1:86400);
rain40(1:n/2) = ts40(1:n/2,Nstations+1);

```

```

rain40(86400-n/2+1:86400) = ts40(86400-n/2+1:86400,Nstations+1);
nogas40 = rain40 - gas40(:,Nstations+1);
scint40 = ts40(:,Nstations+1) - rain40;
tsl4(:,Nstations+1) = fscale(39.5,14,nogas40) + sscale(39.5,14,scint40,0.6,0.6,30)
+ gasl4;
tsl2(:,Nstations+1) = fscale(39.5,12,nogas40) + sscale(39.5,12,scint40,0.6,0.6,30)
+ gasl2;
disp(['Update Matrix: ' num2str(toc)]);
end
save('Final_detailed_stats_results_4stations');

```


Appendix U2

Final statistical analysis with 1 ku band station

```
clear;
load('daysInfo2');
load('inter_session_indices_620');
load('SystemDefinitions');
load('Fwd_ctrl_info','Rs_fwd_ctrl','B_fwd_ctrl');
load('Rtn_ctrl_info','Rs_rtn_ctrl','B_rtn_ctrl');
load('Fwd_traf_info2','Rs_fwd_traf','Rs_fwd_traf_ku','B','B_ku');
load('Rtn_traf_info2','Rs_rtn_traf','Rs_rtn_traf_ku','B_per_channel','B_per_channel_ku');
load('OutboundTraffic_mixed2','CtoN_req','CtoN_req_ku','CtoN_tot','CtoN_tot_ku','duration','marginout');
CNR_req_fwd = CtoN_req;
CNR_req_fwd_ku = CtoN_req_ku;
CNR_ca_fwd = CtoN_tot;
CNR_ca_fwd_ku = CtoN_tot_ku;
duration_fwd = duration;
margin_fwd = marginout;
load('InboundTraffic_mixed2','CtoN_req','CtoN_req_ku','CtoN_tot','CtoN_tot_ku','duration','marginin');
CNR_req_rtn = CtoN_req;
CNR_req_rtn_ku = CtoN_req_ku;
CNR_ca_rtn = CtoN_tot;
CNR_ca_rtn_ku = CtoN_tot_ku;
duration_rtn = duration;
margin_rtn = marginin;
clear CtoN_req CtoN_req_ku CtoN_tot CtoN_tot_ku duration duration_fwd duration_rtn marginin marginout;
load('Final_fixed_margin_CNRS');

% System info
Nstations = find(activity_cdf==0,1)-1;
Phub_traf_V = 5;
Phub_traf_ku = 199;
Phub_ctrl = 1;
Pvsat_traf_V = 0.8;
Pvsat_traf_ku = 2*0.8;
Pvsat_ctrl = 2*0.2;
TransponderBW = 36e6;
Psat_fwd_traf = Psat*B/TransponderBW;
Psat_rtn_traf = Psat*B_per_channel/TransponderBW;
Psat_fwd_traf_ku = Psat*B_ku/TransponderBW;
Psat_rtn_traf_ku = Psat*B_per_channel_ku/TransponderBW;
Psat_fwd_ctrl = Psat*B_fwd_ctrl/TransponderBW;
Psat_rtn_ctrl = Psat*B_rtn_ctrl/TransponderBW;
Psat_fwd_traf_fixed = Psat*B_fwd_fixed/TransponderBW;
Psat_rtn_traf_fixed = Psat*B_per_channel_fixed/TransponderBW;
Tsysca_sat_V = 900;
Tsysca_sat_ku = 500;
Tsysca_ground_V = 350;
Tsysca_ground_ku = 150;
Leoc_hub = 1;
Leoc_vsat = 3;
Lmisc_hub = 1;
Lmisc_vsat = 1;
FMT_CNR_reqs = [8.5 7.9 7.2 6.7 5.8 4.9 4.2 3.7 2.8];
FMT_code_rates = {'6/7','4/5','3/4','2/3','1/2','4/5','3/4','2/3','1/2'};
FMT_bits_per_symbol = [2 2 2 2 1 1 1 1];
no_bursts_per_channel_ku = 1;
Gdif = Grx_hub_ku - Grx_vsat_ku;
scint_margin = 1;
scint_margin_ku = 1;
gas14 = 0.7;
gas12 = 0.5;

% design lpf & hpf
n = 40;
fs = 1;
cutoff = 20e-3;
```

```

Wn = cutoff/(fs/2);
blpf = fir1(n,Wn);
a = 1;

% Initialize working memory
ts40 = zeros(86400,Nstations+1);
ts50 = zeros(86400,Nstations+1);
gas40 = zeros(86400,Nstations+1);
gas50 = zeros(86400,Nstations+1);
ts14 = zeros(86400,Nstations+1);
ts12 = zeros(86400,Nstations+1);
CNR_fwd_down_V = zeros(86400,Nstations);
CNR_rtn_up_V = zeros(86400,Nstations);
CNR_fwd_down_ku = zeros(86400,Nstations);
CNR_rtn_up_ku = zeros(86400,Nstations);
CNR_fwd_down_fixed = zeros(86400,Nstations);
CNR_rtn_up_fixed = zeros(86400,Nstations);
CNR_fwd_V = zeros(86400,Nstations);
CNR_rtn_V = zeros(86400,Nstations);
CNR_fwd_ku = zeros(86400,Nstations);
CNR_rtn_ku = zeros(86400,Nstations);
CNR_fwd_fixed = zeros(86400,Nstations);
CNR_rtn_fixed = zeros(86400,Nstations);
CNR_fwd_pred_V = zeros(86400,Nstations);
CNR_fwd_pred_ku = zeros(86400,Nstations);
active_station_info = zeros(Nstations,1);
errors_rcst_ind = zeros(8,1);
call_count = ones(Nstations_tot,1);
[ len,dum] = size(inter_session_start);
event_count = 0;
channel_type = zeros(86400,Nstations);
csec = 0;
xh = 0:0.001:1;
outages_fading = 0;
outages_fading_fixed = 0;
outages_fading_fixed_nomargin = 0;
outages_noresource = 0;
no_active_station_seconds = 0;
event_no = 0;
station_outage_count = zeros(Nstations_tot,1);
outages_fwd = zeros(Nstations_tot,1);
outages_fwd_fixed = zeros(Nstations_tot,1);
burst_duration_fwd = zeros(1,1001);
burst_duration_fwd_req = zeros(1,1001);
outages_fixed = zeros(Nstations_tot,1);
outages_fixed_nomargin = zeros(Nstations_tot,1);
traf_fmt_ratio = zeros(1,101);

% Create lookup table for dNrain, indexed by attenuation
dNrain_ku = zeros(101,1);
CNR_reduction_ku = zeros(101,1);
CNR_reduction_hub_ku = zeros(101,1);
Taca_ku = 275*(1-10^-(gas12/10));
for att = 0:100
    Tarain_ku = 275*(1-10^-( (gas12+att)/10));
    dTsys_ku = Tarain_ku - Taca_ku;
    Tsysrain_ku = Tsysca_ground_ku + dTsys_ku;
    dNrain_ku(att+1) = 10*log10(Tsysrain_ku/Tsysca_ground_ku);
    CNR_reduction_ku(att+1) = att + dNrain_ku(att+1);
    CNR_reduction_hub_ku(att+1) = fscale(12,14,att) + att + dNrain_ku(att+1);
end
Taca_V = 275*(1-10^-(1/10));

% Simulation info
no_days = length(daysInfo);

for c = 1:Nstations+1
    date = daysInfo(c).date;
    dummycell = load(['Data3/S40_' date '.mat']);
    eval(['ts40(:,c) = dummycell.S40_' date '(:,2);']);
    eval(['gas40(:,c) = dummycell.S40_' date '(:,2) - dummycell.S40_' date '(:,1);']);
    dummycell = load(['Data3/S50_' date '.mat']);
    eval(['ts50(:,c) = dummycell.S50_' date '(:,2);']);
    eval(['gas50(:,c) = dummycell.S50_' date '(:,2) - dummycell.S50_' date '(:,1);']);
    [rain40,zr] = filter(blpf,a,ts40(:,c),ts40(1,c)*ones(n,1));
    rain40(1:86400-n/2) = rain40(n/2+1:86400);
    rain40(1:n/2) = ts40(1:n/2,c);

```

```

rain40(86400-n/2+1:86400) = ts40(86400-n/2+1:86400,c);
nogas40 = rain40 - gas40(:,c);
scint40 = ts40(:,c) - rain40;
tsl4(:,c) = fscale(39.5,14,nogas40) + sscale(39.5,14,scint40,0.6,0.6,30) + gas14;
tsl2(:,c) = fscale(39.5,12,nogas40) + sscale(39.5,12,scint40,0.6,0.6,30) + gas12;
end

for day = 1:no_days % Loop over all days in daysinfo
    event_time = zeros(Nstations,1);
    event_flag = zeros(Nstations,1);
    min_CNR = 20*ones(Nstations,1);
    tic;
    disp(['Doing day ' int2str(day)]);
    % Change single site time series to joint site time series
    timeshift = 14.5*60;
    for c = 1:86400-timeshift
        tsl2(c,1) = min(tsl2(c,1),tsl2(c+timeshift,1));
        tsl4(c,1) = min(tsl4(c,1),tsl4(c+timeshift,1));
        ts40(c,1) = min(ts40(c,1),ts40(c+timeshift,1));
        ts50(c,1) = min(ts50(c,1),ts50(c+timeshift,1));
    end
    for c = 86400-timeshift+1:86400
        tsl2(c,1) = min(tsl2(c,1),tsl2(c-86400+timeshift,2));
        tsl4(c,1) = min(tsl4(c,1),tsl4(c-86400+timeshift,2));
        ts40(c,1) = min(ts40(c,1),ts40(c-86400+timeshift,2));
        ts50(c,1) = min(ts50(c,1),ts50(c-86400+timeshift,2));
    end
    disp(['Site Diversity: ' num2str(toc)]);
    tic;
    % Calculate hub side CNRs
    CNR_fwd_up_V =
att2cnr(ts50(:,1),gas50(:,1),Gtx_hub_V+10*log10(Phub_traf_V),Grx_satellite_V,Tsysca_sat_V,ulink_freq,Rs_fwd_traf,Leoc_hub,Lmisc_hub,0);
    CNR_rtn_down_V =
att2cnr(ts40(:,1),gas40(:,1),Gtx_satellite_V+10*log10(Psat_rtn_traf),Grx_hub_V,Tsysca_ground_V,dlink_freq,Rs_rtn_traf,Leoc_hub,Lmisc_hub,1);
    CNR_fwd_up_ku =
att2cnr(tsl4(:,1),gas14*ones(86400,1),Gtx_hub_ku+10*log10(Phub_traf_ku),Grx_satellite_ku,Tsysca_sat_ku,ulink_freq_ku,Rs_fwd_traf_ku,Leoc_hub,Lmisc_hub,0);
    CNR_rtn_down_ku =
att2cnr(tsl2(:,1),gas12*ones(86400,1),Gtx_satellite_ku+10*log10(Psat_rtn_traf_ku),Grx_hub_ku,Tsysca_ground_ku,dlink_freq_ku,Rs_rtn_traf_ku,Leoc_hub,Lmisc_hub,1);
    CNR_fwd_up_fixed =
att2cnr(ts50(:,1),gas50(:,1),Gtx_hub_V+10*log10(Phub_traf_V),Grx_satellite_V,Tsysca_sat_V,ulink_freq,Rs_fwd_fixed,Leoc_hub,Lmisc_hub,0);
    CNR_rtn_down_fixed =
att2cnr(ts40(:,1),gas40(:,1),Gtx_satellite_V+10*log10(Psat_rtn_traf),Grx_hub_V,Tsysca_ground_V,dlink_freq,Rs_rtn_fixed,Leoc_hub,Lmisc_hub,1);
    CNR_fwd_up_ku0 =
att2cnr(gas14,gas14,Gtx_hub_ku+10*log10(Phub_traf_ku),Grx_satellite_ku,Tsysca_sat_ku,ulink_freq_ku,Rs_fwd_traf_ku,Leoc_hub,Lmisc_hub,0);
    CNR_fwd_up_V0 =
att2cnr(gas50(:,1),gas50(:,1),Gtx_hub_V+10*log10(Phub_traf_V),Grx_satellite_V,Tsysca_sat_V,ulink_freq,Rs_fwd_traf,Leoc_hub,Lmisc_hub,0);

    % Calculate forward downlink for the hub.
    CNR_fwd_down_hub_ku =
att2cnr(tsl2(:,1),gas12*ones(86400,1),Gtx_satellite_ku+10*log10(Psat_fwd_traf_ku),Grx_hub_ku,Tsysca_ground_ku,dlink_freq_ku,Rs_fwd_traf_ku,Leoc_hub,Lmisc_hub,1);
    CNR_fwd_down_hub_ku0 =
att2cnr(gas12,gas12,Gtx_satellite_ku+10*log10(Psat_fwd_traf_ku),Grx_hub_ku,Tsysca_ground_ku,dlink_freq_ku,Rs_fwd_traf_ku,Leoc_hub,Lmisc_hub,1);
    CNR_fwd_hub_0 = 10*log10(1./((1./(10.^(CNR_fwd_up_ku0/10))) +
(1./10.^(CNR_fwd_down_hub_ku0/10))));
    CNR_fwd_hub_ku = 10*log10(1./((1./(10.^(CNR_fwd_up_ku/10))) +
(1./10.^(CNR_fwd_down_hub_ku/10))));
    CNR_fwd_pred_hub_ku = rtadalinemex(CNR_fwd_hub_ku,3,1,1e-6,[0.3 0.3
0.4],CNR_fwd_hub_ku(1));

    AttNoise_ku = CNR_fwd_hub_0 - CNR_fwd_pred_hub_ku;
    Att_ku = interp1(CNR_reduction_hub_ku,(0:100)',AttNoise_ku,'linear','extrap');
    Att_V = fscale(12,50,Att_ku);

    % Adjust overall CNR based on power control
    CNR_fwd_up_V_pctrl = CNR_fwd_up_V;
    for c = 1:86399

```

```

        if Att_V(c) > 6
            CNR_fwd_up_V_pctrl(c+1) = CNR_fwd_up_V(c+1) + 6;
        elseif Att_V(c) > 3
            CNR_fwd_up_V_pctrl(c+1) = CNR_fwd_up_V(c+1) + 3;
        end
    end
    disp(['Power Control: ' num2str(toc)]);
    tic;

    % Do vsat side CNR calculations & total CNR calculations with power control
    for c = 1:Nstations
        CNR_fwd_down_V(:,c) =
att2cnr(ts40(:,c+1),gas40(:,c+1),Gtx_satellite_V+10*log10(Psat_fwd_traf),Grx_vsat_V,Ts
ysca_ground_V,dlink_freq,Rs_fwd_traf,Leoc_vsat,Lmisc_vsat,1);
        CNR_rtn_up_V(:,c) =
att2cnr(ts50(:,c+1),gas50(:,c+1),Gtx_vsat_V+10*log10(Pvsat_traf_V),Grx_satellite_V,Tsy
sca_sat_V,ulink_freq,Rs_rtn_traf,Leoc_vsat,Lmisc_vsat,0);
        CNR_fwd_down_ku(:,c) =
att2cnr(tsl2(:,c+1),gas12*ones(86400,1),Gtx_satellite_ku+10*log10(Psat_fwd_traf_ku),Gr
x_vsat_ku,Tsysca_ground_ku,dlink_freq_ku,Rs_fwd_traf_ku,Leoc_vsat,Lmisc_vsat,1);
        CNR_rtn_up_ku(:,c) =
att2cnr(tsl4(:,c+1),gas14*ones(86400,1),Gtx_vsat_ku+10*log10(Pvsat_traf_ku),Grx_satell
ite_ku,Tsysca_sat_ku,ulink_freq_ku,Rs_rtn_traf_ku,Leoc_vsat,Lmisc_vsat,0);
        CNR_fwd_down_fixed(:,c) =
att2cnr(ts40(:,c+1),gas40(:,c+1),Gtx_satellite_V+10*log10(Psat_fwd_traf),Grx_vsat_V,Ts
ysca_ground_V,dlink_freq,Rs_fwd_fixed,Leoc_vsat,Lmisc_vsat,1);
        CNR_rtn_up_fixed(:,c) =
att2cnr(ts50(:,c+1),gas50(:,c+1),Gtx_vsat_V+10*log10(Pvsat_traf_V),Grx_satellite_V,Tsy
sca_sat_V,ulink_freq,Rs_rtn_fixed,Leoc_vsat,Lmisc_vsat,0);
        CNR_fwd_down_ku0(:,c) =
att2cnr(gas12,gas12,Gtx_satellite_ku+10*log10(Psat_fwd_traf_ku),Grx_vsat_ku,Tsysca_gro
und_ku,dlink_freq_ku,Rs_fwd_traf_ku,Leoc_vsat,Lmisc_vsat,1);
        CNR_fwd_down_V0(:,c) =
att2cnr(gas40(:,c+1),gas40(:,c+1),Gtx_satellite_V+10*log10(Psat_fwd_traf),Grx_vsat_V,T
sysca_ground_V,dlink_freq,Rs_fwd_traf,Leoc_vsat,Lmisc_vsat,1);

        CNR_fwd_V(:,c) = 10*log10(1./((1./(10.^(CNR_fwd_up_V_pctrl/10))) +
(1./10.^(CNR_fwd_down_V(:,c)/10))));
        CNR_rtn_V(:,c) = 10*log10(1./((1./(10.^(CNR_rtn_up_V(:,c)/10))) +
(1./10.^(CNR_rtn_down_V/10))));
        CNR_fwd_ku(:,c) = 10*log10(1./((1./(10.^(CNR_fwd_up_ku/10))) +
(1./10.^(CNR_fwd_down_ku(:,c)/10))));
        CNR_rtn_ku(:,c) = 10*log10(1./((1./(10.^(CNR_rtn_up_ku(:,c)/10))) +
(1./10.^(CNR_rtn_down_ku/10))));
        CNR_tmp = 10*log10(1./((1./(10.^(CNR_fwd_up_fixed/10))) +
(1./10.^(CNR_fwd_down_fixed(:,c)/10))));
        CNR_fwd_fixed(:,c) = filter(blpf,a,CNR_tmp);
        CNR_fwd_fixed(1:86400-n/2) = CNR_fwd_fixed(n/2+1:86400);
        CNR_fwd_fixed(86400-n/2+1:86400) = CNR_tmp(86400-n/2+1:86400);
        CNR_tmp = 10*log10(1./((1./(10.^(CNR_rtn_up_fixed(:,c)/10))) +
(1./10.^(CNR_rtn_down_fixed/10))));
        CNR_rtn_fixed(:,c) = filter(blpf,a,CNR_tmp);
        CNR_rtn_fixed(1:86400-n/2) = CNR_rtn_fixed(n/2+1:86400);
        CNR_rtn_fixed(86400-n/2+1:86400) = CNR_tmp(86400-n/2+1:86400);

        CNR_fwd_pred_ku(:,c) = rtadalinemex(CNR_fwd_ku(:,c),3,4,1e-6,[0.3 0.3
0.4],CNR_fwd_ku(1,c));
        CNR_fwd_pred_V(:,c) = rtadalinemex(CNR_fwd_V(:,c),3,4,1e-6,[0.3 0.3
0.4],CNR_fwd_V(1,c));
    end
    disp(['VSAT CNRs & Predictions: ' num2str(toc)]);

    tic;
    for dsec = 1:86400
        csec = csec+1;
        if dsec==1
            esec = 1;
        else
            esec = dsec - 1;
        end
        if dsec < 4
            fsec = 4;
        else
            fsec = dsec;
        end
        % Assign active station with CNR from one of the active channels
        for c = 1:Nstations_tot

```

```

        if csec == inter_session_start(call_count(c),c)
            active_station_info(find(active_station_info==0,1)) = c;
        end
        if csec == inter_session_end(call_count(c),c)
            active_station_info(find(active_station_info==c,1)) = 0;
            if call_count(c) < len
                call_count(c) = call_count(c) + 1;
            end
        end
    end
end

active_index = find(active_station_info~=0);
active_sum = length(active_index);
if active_sum
    no_active_station_seconds = no_active_station_seconds + active_sum;
    % Determine which stations are using ku band
    ind_ku = linspace(1,no_bursts_per_channel_ku,no_bursts_per_channel_ku);
    val_ku = 10000*ones(no_bursts_per_channel_ku,1);
    outage_vector = zeros(active_sum,1);
    for c = 1:active_sum
        mc = 1;
        notfilled = 1;
        if any(CNR_fwd_pred_ku(fsec-3:fsec,active_index(c)) > CNR_req_fwd_ku)
            while notfilled && mc <= no_bursts_per_channel_ku
                if mean(CNR_fwd_pred_V(fsec-3:fsec,active_index(c))) <
val_ku(mc)

val_ku(mc+1:no_bursts_per_channel_ku)=val_ku(mc:no_bursts_per_channel_ku-1);
                val_ku(mc) = mean(CNR_fwd_pred_ku(fsec-
3:fsec,active_index(c)));

ind_ku(mc+1:no_bursts_per_channel_ku)=ind_ku(mc:no_bursts_per_channel_ku-1);
                ind_ku(mc) = c;
                notfilled = 0;
            end
            mc = mc + 1;
        end
    end
    end
    outagestmp_fading = zeros(Nstations,1);
    outagestmp_fading_fixed = zeros(Nstations,1);
    outagestmp_fading_fixed_nomargin = zeros(Nstations,1);
    outagestmp_rtn_fixed = zeros(Nstations,1);
    burst_duration_fwd_temp = zeros(Nstations,1);
    burst_duration_rtn_temp = zeros(Nstations,1);
    for c = 1:active_sum
        if ~event_flag(active_index(c)) &&
CNR_fwd_pred_ku(dsec,active_index(c)) < CNR_req_fwd_ku
            event_flag(active_index(c)) = 1;
        end
        if event_flag(active_index(c))
            outagestmp_fading(active_index(c)) = 1;
            outages_fwd(active_station_info(active_index(c))) =
outages_fwd(active_station_info(active_index(c))) + 1;
            if all(CNR_fwd_pred_ku(fsec-3:fsec,active_index(c)) >
CNR_req_fwd_ku)
                event_flag(active_index(c)) = 0;
            end
        end

        if CNR_fwd_fixed(dsec,active_index(c)) < CNR_req_fwd_fixed;
            outagestmp_fading_fixed_nomargin(active_index(c)) = 1;
            outages_fixed_nomargin(active_station_info(active_index(c))) =
outages_fixed_nomargin(active_station_info(active_index(c))) + 1;
        end
        % Find burst duration for each station using V-band
        if ~any(ind_ku == c)
            if all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(1) + scint_margin)
                burst_duration_fwd_temp(active_index(c)) =
duration_fwd(1);
            elseif all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(2) + scint_margin)
                burst_duration_fwd_temp(active_index(c)) =
duration_fwd(2);
            elseif all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(3) + scint_margin)

```

```

        burst_duration_fwd_temp(active_index(c)) =
duration_fwd(3);
        elseif all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(4) + scint_margin)
        burst_duration_fwd_temp(active_index(c)) =
duration_fwd(4);
        elseif all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(5) + scint_margin)
        burst_duration_fwd_temp(active_index(c)) =
duration_fwd(5);
        elseif all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(6) + scint_margin)
        burst_duration_fwd_temp(active_index(c)) =
duration_fwd(6);
        elseif all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(7) + scint_margin)
        burst_duration_fwd_temp(active_index(c)) =
duration_fwd(7);
        elseif all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(8) + scint_margin)
        burst_duration_fwd_temp(active_index(c)) =
duration_fwd(8);
        elseif all(CNR_fwd_pred_V(fsec-3:fsec,active_index(c)) >
CNR_req_fwd - margins_fwd(9) + scint_margin)
        burst_duration_fwd_temp(active_index(c)) =
duration_fwd(9);
    end
end

end
outages_fading = outages_fading + sum(outagestmp_fading);
outages_fading_fixed = outages_fading_fixed +
sum(outagestmp_fading_fixed);
outages_fading_fixed_nomargin = outages_fading_fixed_nomargin +
sum(outagestmp_fading_fixed_nomargin);

burst_duration_fwd_all_stations = sum(burst_duration_fwd_temp);
active_sum_modified = active_sum;
burst_duration_fwd_req = burst_duration_fwd_req +
hist(sum(burst_duration_fwd_temp),xh);
while burst_duration_fwd_all_stations > 20e-3
    [max_dur,max_ind] = max(burst_duration_fwd_temp);
    burst_duration_fwd_temp(max_ind) = 0;
    burst_duration_fwd_all_stations = burst_duration_fwd_all_stations -
max_dur;
    outages_noresource = outages_noresource + 1;
    active_sum_modified = active_sum_modified - 1;
end
burst_duration_fwd = burst_duration_fwd +
hist(sum(burst_duration_fwd_temp),xh);
dur_ratio = (burst_duration_fwd_all_stations -
active_sum_modified*duration_fwd(1))/burst_duration_fwd_all_stations;
traf_fmt_ratio = traf_fmt_ratio + hist(dur_ratio,0:0.001:0.1);
end
end
disp(['Stats: ' num2str(toc)]);
tic;
% Update matrix
ts12(:,1:Nstations) = ts12(:,2:Nstations+1);
ts14(:,1:Nstations) = ts14(:,2:Nstations+1);
ts40(:,1:Nstations) = ts40(:,2:Nstations+1);
ts50(:,1:Nstations) = ts50(:,2:Nstations+1);
gas40(:,1:Nstations) = gas40(:,2:Nstations+1);
gas50(:,1:Nstations) = gas50(:,2:Nstations+1);
date = daysInfo(rem(day+Nstations,no_days)+1).date;
dummycell = load(['Data3/S40_' date '.mat']);
eval(['ts40(:,Nstations+1) = dummycell.S40_' date '(:,2);']);
eval(['gas40(:,Nstations+1) = dummycell.S40_' date '(:,2) - dummycell.S40_' date
'(:,1);']);
dummycell = load(['Data3/S50_' date '.mat']);
eval(['ts50(:,Nstations+1) = dummycell.S50_' date '(:,2);']);
eval(['gas50(:,Nstations+1) = dummycell.S50_' date '(:,2) - dummycell.S50_' date
'(:,1);']);
[rain40,zr] = filter(blpf,a,ts40(:,Nstations+1),ts40(1,Nstations+1)*ones(n,1));
rain40(1:86400-n/2) = rain40(n/2+1:86400);
rain40(1:n/2) = ts40(1:n/2,Nstations+1);
rain40(86400-n/2+1:86400) = ts40(86400-n/2+1:86400,Nstations+1);

```

```

    nogas40 = rain40 - gas40(:,Nstations+1);
    scint40 = ts40(:,Nstations+1) - rain40;
    tsl4(:,Nstations+1) = fscale(39.5,14,nogas40) + sscale(39.5,14,scint40,0.6,0.6,30)
+ gas14;
    tsl2(:,Nstations+1) = fscale(39.5,12,nogas40) + sscale(39.5,12,scint40,0.6,0.6,30)
+ gas12;
    disp(['Update Matrix: ' num2str(toc)]);
end
save('Final_detailed_stats_results_1station');

```


Appendix V1

Final event analysis with 4 ku band stations

```
clear;
load('daysInfo2');
load('inter_session_indices_620');
% load('inter_session_indices_formatted');
load('SystemDefinitions');
load('Fwd_ctrl_info','Rs_fwd_ctrl','B_fwd_ctrl');
load('Rtn_ctrl_info','Rs_rtn_ctrl','B_rtn_ctrl');
load('Fwd_traf_info','Rs_fwd_traf','Rs_fwd_traf_ku','B','B_ku');
load('Rtn_traf_info','Rs_rtn_traf','Rs_rtn_traf_ku','B_per_channel','B_per_channel_ku');
load('OutboundTraffic_mixed','CtoN_req','CtoN_req_ku','CtoN_tot','CtoN_tot_ku','durout',
    'marginout');
CNR_req_fwd = CtoN_req;
CNR_req_fwd_ku = CtoN_req_ku;
CNR_ca_fwd = CtoN_tot;
CNR_ca_fwd_ku = CtoN_tot_ku;
duration_fwd = durout;
margins_fwd = marginout;
load('InboundTraffic_mixed','CtoN_req','CtoN_req_ku','CtoN_tot','CtoN_tot_ku','durin',
    'marginin');
CNR_req_rtn = CtoN_req;
CNR_req_rtn_ku = CtoN_req_ku;
CNR_ca_rtn = CtoN_tot;
CNR_ca_rtn_ku = CtoN_tot_ku;
duration_rtn = durin;
margins_rtn = marginin;
clear CtoN_req CtoN_req_ku CtoN_tot CtoN_tot_ku durin durout marginin marginout;
load('Final_detailed_find_events_results_620_2','event_list');

event_no = 92;
day = event_list(event_no).day;
start_time = event_list(event_no).start;
end_time = event_list(event_no).end;
spec_stat = event_list(event_no).station;
spec_series = event_list(event_no).series;

% System info
Nstations = find(activity_cdf==0,1)-1;
Phub_traf_V = 5;
Phub_traf_ku = 199;
Phub_ctrl = 1;
Pvsat_traf_V = 0.8;
Pvsat_traf_ku = 2*0.8;
Pvsat_ctrl = 2*0.2;
TransponderBW = 36e6;
Psat_fwd_traf = Psat*B/TransponderBW;
Psat_rtn_traf = Psat*B_per_channel/TransponderBW;
Psat_fwd_traf_ku = Psat*B_ku/TransponderBW;
Psat_rtn_traf_ku = Psat*B_per_channel_ku/TransponderBW;
Psat_fwd_ctrl = Psat*B_fwd_ctrl/TransponderBW;
Psat_rtn_ctrl = Psat*B_rtn_ctrl/TransponderBW;
Tsysca_sat_V = 900;
Tsysca_sat_ku = 500;
Tsysca_ground_V = 350;
Tsysca_ground_ku = 150;
Leoc_hub = 1;
Leoc_vsat = 3;
Lmisc_hub = 1;
Lmisc_vsat = 1;
FMT_CNR_reqs = [8.5 7.9 7.2 6.7 5.8 4.9 4.2 3.7 2.8];
FMT_code_rates = {'6/7','4/5','3/4','2/3','1/2','4/5','3/4','2/3','1/2'};
FMT_bits_per_symbol = [2 2 2 2 1 1 1 1];
no_bursts_per_channel_ku = 4;
scint_margin = 1;

% design lpf & hpf
n = 40;
```



```

fs = 1;
cutoff = 20e-3;
Wn = cutoff/(fs/2);
blpf = fir1(n,Wn);
a = 1;

% Initialize working memory
ts40 = zeros(86400,1);
ts50 = zeros(86400,1);
gas40 = zeros(86400,1);
gas50 = zeros(86400,1);
CNR_fwd_down_V = zeros(86400,Nstations);
CNR_rtn_up_V = zeros(86400,Nstations);
CNR_fwd_down_ku = zeros(86400,Nstations);
CNR_rtn_up_ku = zeros(86400,Nstations);
CNR_fwd_down_ctrl = zeros(86400,Nstations);
CNR_rtn_up_ctrl = zeros(86400,Nstations);
CNR_fwd_V = zeros(86400,Nstations);
CNR_rtn_V = zeros(86400,Nstations);
CNR_fwd_pred_ku = zeros(86400,Nstations);
active_station_info = zeros(Nstations,1);
biterrors_fwd_ctrl_nccflc = zeros(Nstations_tot,1);
biterrors_fwd_ctrl_eack = zeros(Nstations_tot,1);
biterrors_rtn_ctrl_rcst = zeros(Nstations_tot,1);
biterrors_rtn_ctrl_ecs = zeros(Nstations_tot,1);
frameerrors_fwd_ctrl_nccflc = zeros(Nstations_tot,1);
frameerrors_fwd_ctrl_eack = zeros(Nstations_tot,1);
frameerrors_rtn_ctrl_rcst = zeros(Nstations_tot,1);
frameerrors_rtn_ctrl_ecs = zeros(Nstations_tot,1);
biterrors_fwd_traf = zeros(Nstations_tot,1);
biterrors_rtn_traf = zeros(Nstations_tot,1);
frameerrors_fwd_traf = zeros(Nstations_tot,1);
frameerrors_rtn_traf = zeros(Nstations_tot,1);
biterrors_fwd_traf_specific = zeros(86400,1);
biterrors_rtn_traf_specific = zeros(86400,1);
biterrors_fwd_ctrl_specific = zeros(86400,1);
biterrors_rtn_ctrl_specific = zeros(86400,1);
FMT_type = ones(86400,58);
% channel_type = zeros(86400,Nstations);
errors_rcst_ind = zeros(8,1);
call_count = ones(Nstations_tot,1);
Nconc = zeros(Nstations,1);
[ len,dum] = size(inter_session_start);
const_BPSK = [1 -1];
const_QPSK = [1;j;-j;-1];
pwr_BPSK = sum(abs(const_BPSK).^2)/length(const_BPSK);
pwr_QPSK = sum(abs(const_QPSK).^2)/length(const_QPSK);
conv_trel = poly2trellis(7,[171,133]);

% Do hub side CNR calculations
date = daysInfo(day).date;
dummycell = load(['Data3/S40_' date '.mat']);
eval(['ts40 = dummycell.S40_' date '(:,2);']);
eval(['gas40 = dummycell.S40_' date '(:,2) - dummycell.S40_' date '(:,1);']);
dummycell = load(['Data3/S50_' date '.mat']);
eval(['ts50 = dummycell.S50_' date '(:,2);']);
eval(['gas50 = dummycell.S50_' date '(:,2) - dummycell.S50_' date '(:,1);']);
% Apply timeshift for site diversity
timeshift = 14.5*60;
for c = 1:86400-timeshift
    ts40(c,1) = min(ts40(c,1),ts40(c+timeshift,1));
    ts50(c,1) = min(ts50(c,1),ts50(c+timeshift,1));
end
[rain40,zr] = filter(blpf,a,ts40,ts40(1)*ones(n,1));
rain40(1:86400-n/2) = rain40(n/2+1:86400);
rain40(1:n/2) = ts40(1:n/2);
rain40(86400-n/2+1:86400) = ts40(86400-n/2+1:86400);
nogas40 = rain40 - gas40;
scint40 = ts40 - rain40;
gas14 = 0.7;
gas12 = 0.5;
ts14 = fscale(39.5,14,nogas40) + sscale(39.5,14,scint40,0.6,0.6,30) + gas14;
ts12 = fscale(39.5,12,nogas40) + sscale(39.5,12,scint40,0.6,0.6,30) + gas12;
CNR_fwd_up_V =
att2cnr(ts50,gas50,Gtx_hub_V+10*log10(Phub_traf_V),Grx_satellite_V,Tsysca_sat_V,ulink_
freq,Rs_fwd_traf,Leoc_hub,Lmisc_hub,0);

```

```

CNR_rtn_down_V =
att2cnr(ts40,gas40,Gtx_satellite_V+10*log10(Psat_rtn_traf),Grx_hub_V,Tsysca_ground_V,d
link_freq,Rs_rtn_traf,Leoc_hub,Lmisc_hub,1);
CNR_fwd_up_ku =
att2cnr(ts14,gas14*ones(86400,1),Gtx_hub_ku+10*log10(Phub_traf_ku),Grx_satellite_ku,Ts
ysca_sat_ku,ulink_freq_ku,Rs_fwd_traf_ku,Leoc_hub,Lmisc_hub,0);
CNR_rtn_down_ku =
att2cnr(ts12,gas12*ones(86400,1),Gtx_satellite_ku+10*log10(Psat_rtn_traf_ku),Grx_hub_k
u,Tsysca_ground_ku,dlink_freq_ku,Rs_rtn_traf_ku,Leoc_hub,Lmisc_hub,1);
CNR_fwd_up_ctrl =
att2cnr(ts14,gas14*ones(86400,1),Gtx_hub_ku+10*log10(Phub_ctrl),Grx_satellite_ku,Tsysc
a_sat_ku,ulink_freq_ku,Rs_fwd_ctrl,Leoc_hub,Lmisc_hub,0);
CNR_rtn_down_ctrl =
att2cnr(ts12,gas12*ones(86400,1),Gtx_satellite_ku+10*log10(Psat_rtn_ctrl),Grx_hub_ku,T
sysca_ground_ku,dlink_freq_ku,Rs_rtn_ctrl,Leoc_hub,Lmisc_hub,1);
CNR_fwd_up_ku0 =
att2cnr(gas14,gas14,Gtx_hub_ku+10*log10(Phub_traf_ku),Grx_satellite_ku,Tsysca_sat_ku,u
link_freq_ku,Rs_fwd_traf_ku,Leoc_hub,Lmisc_hub,0);
CNR_fwd_up_V0 =
att2cnr(gas50(:,1),gas50(:,1),Gtx_hub_V+10*log10(Phub_traf_V),Grx_satellite_V,Tsysca_s
at_V,ulink_freq,Rs_fwd_traf,Leoc_hub,Lmisc_hub,0);

% Create lookup table for dNrain, indexed by attenuation
dNrain_ku = zeros(101,1);
CNR_reduction_ku = zeros(101,1);
CNR_reduction_hub_ku = zeros(101,1);
Taca_ku = 275*(1-10^(-(gas12/10)));
for att = 0:100
    Tarain_ku = 275*(1-10^(-(gas12+att)/10));
    dTsys_ku = Tarain_ku - Taca_ku;
    Tsyrain_ku = Tsysca_ground_ku + dTsys_ku;
    dNrain_ku(att+1) = 10*log10(Tsyrain_ku/Tsysca_ground_ku);
    CNR_reduction_ku(att+1) = att + dNrain_ku(att+1);
    CNR_reduction_hub_ku(att+1) = fscale(12,14,att) + att + dNrain_ku(att+1);
end
Taca_V = 275*(1-10^(-(1/10)));

% Calculate forward downlink for the hub.
CNR_fwd_down_hub_ku =
att2cnr(ts12(:,1),gas12*ones(86400,1),Gtx_satellite_ku+10*log10(Psat_fwd_traf_ku),Grx_
hub_ku,Tsysca_ground_ku,dlink_freq_ku,Rs_fwd_traf_ku,Leoc_hub,Lmisc_hub,1);
CNR_fwd_down_hub_ku0 =
att2cnr(gas12,gas12,Gtx_satellite_ku+10*log10(Psat_fwd_traf_ku),Grx_hub_ku,Tsysca_grou
nd_ku,dlink_freq_ku,Rs_fwd_traf_ku,Leoc_hub,Lmisc_hub,1);
CNR_fwd_hub_0 = 10*log10(1./((1./((10.^(CNR_fwd_up_ku0/10))) +
(1./10.^(CNR_fwd_down_hub_ku0/10)))));
CNR_fwd_hub_ku = 10*log10(1./((1./((10.^(CNR_fwd_up_ku/10))) +
(1./10.^(CNR_fwd_down_hub_ku/10)))));
CNR_fwd_pred_hub_ku = rtadalinemex(CNR_fwd_hub_ku,3,1,1e-6,[0.3 0.3
0.4],CNR_fwd_hub_ku(1));

AttNoise_ku = CNR_fwd_hub_0 - CNR_fwd_pred_hub_ku;
Att_ku = interp1(CNR_reduction_hub_ku,(0:100)',AttNoise_ku,'linear','extrap');
Att_V = fscale(12,50,Att_ku);

disp('Done hub side calculations');

% Adjust overall CNR based on power control
CNR_fwd_up_V_pctrl = CNR_fwd_up_V;
for c = 1:86399
    if Att_V(c) > 6
        CNR_fwd_up_V_pctrl(c+1) = CNR_fwd_up_V(c+1) + 6;
    elseif Att_V(c) > 3
        CNR_fwd_up_V_pctrl(c+1) = CNR_fwd_up_V(c+1) + 3;
    end
end

% Do vsat side CNR calculations & total CNR calculations with power control
for c = 1:Nstations
    varday = rem(day+c,length(daysInfo));
    if varday == 0
        varday = length(daysInfo);
    end
    date = daysInfo(varday).date;
    dummycell = load(['Data3/S40_' date '.mat']);
    eval(['ts40 = dummycell.S40_' date '(:,2);']);

```

```

eval(['gas40 = dummycell.S40_' date '(:,2) - dummycell.S40_' date '(:,1);']);
dummycell = load(['Data3/S50_' date '.mat']);
eval(['ts50 = dummycell.S50_' date '(:,2);']);
eval(['gas50 = dummycell.S50_' date '(:,2) - dummycell.S50_' date '(:,1);']);
[rain40,zr] = filter(blpf,a,ts40,ts40(1)*ones(n,1));
rain40(1:86400-n/2) = rain40(n/2+1:86400);
rain40(86400-n/2+1:86400) = zr(1:n/2);
nogas40 = rain40 - gas40;
scint40 = ts40 - rain40;
tsl4 = fscale(39.5,14,nogas40) + sscale(39.5,14,scint40,0.6,0.6,30) + gas14;
tsl2 = fscale(39.5,12,nogas40) + sscale(39.5,12,scint40,0.6,0.6,30) + gas12;
CNR_fwd_down_V(:,c) =
att2cnr(ts40,gas40,Gtx_satellite_V+10*log10(Psat_fwd_traf),Grx_vsat_V,Tsysca_ground_V,
dlink_freq,Rs_fwd_traf,Leoc_vsat,Lmisc_vsat,1);
CNR_rtn_up_V(:,c) =
att2cnr(ts50,gas50,Gtx_vsat_V+10*log10(Pvsat_traf_V),Grx_satellite_V,Tsysca_sat_V,ulink
k_freq,Rs_rtn_traf,Leoc_vsat,Lmisc_vsat,0);
CNR_fwd_down_ku(:,c) =
att2cnr(tsl2,gas12*ones(86400,1),Gtx_satellite_ku+10*log10(Psat_fwd_traf_ku),Grx_vsat_
ku,Tsysca_ground_ku,dlink_freq_ku,Rs_fwd_traf_ku,Leoc_vsat,Lmisc_vsat,1);
CNR_rtn_up_ku(:,c) =
att2cnr(tsl4,gas14*ones(86400,1),Gtx_vsat_ku+10*log10(Pvsat_traf_ku),Grx_satellite_ku,
Tsysca_sat_ku,ulink_freq_ku,Rs_rtn_traf_ku,Leoc_vsat,Lmisc_vsat,0);
CNR_fwd_down_ctrl(:,c) =
att2cnr(tsl2,gas12*ones(86400,1),Gtx_satellite_ku+10*log10(Psat_fwd_ctrl),Grx_vsat_ku,
Tsysca_ground_ku,dlink_freq_ku,Rs_fwd_ctrl,Leoc_vsat,Lmisc_vsat,1);
CNR_rtn_up_ctrl(:,c) =
att2cnr(tsl4,gas14*ones(86400,1),Gtx_vsat_ku+10*log10(Pvsat_ctrl),Grx_satellite_ku,Tsy
sca_sat_ku,ulink_freq_ku,Rs_rtn_ctrl,Leoc_vsat,Lmisc_vsat,0);
CNR_fwd_down_ku0(:,c) =
att2cnr(gas12,gas12,Gtx_satellite_ku+10*log10(Psat_fwd_traf_ku),Grx_vsat_ku,Tsysca_gro
und_ku,dlink_freq_ku,Rs_fwd_traf_ku,Leoc_vsat,Lmisc_vsat,1);
CNR_fwd_down_V0(:,c) =
att2cnr(gas40,gas40,Gtx_satellite_V+10*log10(Psat_fwd_traf),Grx_vsat_V,Tsysca_ground_V
,dlink_freq,Rs_fwd_traf,Leoc_vsat,Lmisc_vsat,1);

CNR_fwd_V(:,c) = 10*log10(1./((1./(10.^(CNR_fwd_up_V_pctrl/10))) +
(1./10.^(CNR_fwd_down_V(:,c)/10))));
CNR_rtn_V(:,c) = 10*log10(1./((1./(10.^(CNR_rtn_up_V(:,c)/10))) +
(1./10.^(CNR_rtn_down_V/10))));
CNR_fwd_ku(:,c) = 10*log10(1./((1./(10.^(CNR_fwd_up_ku/10))) +
(1./10.^(CNR_fwd_down_ku(:,c)/10))));
CNR_rtn_ku(:,c) = 10*log10(1./((1./(10.^(CNR_rtn_up_ku(:,c)/10))) +
(1./10.^(CNR_rtn_down_ku/10))));
CNR_fwd_ku0(:,c) = 10*log10(1./((1./(10.^(CNR_fwd_up_ku0/10))) +
(1./10.^(CNR_fwd_down_ku0(:,c)/10))));
CNR_fwd_V0(:,c) = 10*log10(1./((1./(10.^(CNR_fwd_up_V0/10))) +
(1./10.^(CNR_fwd_down_V0(:,c)/10))));

CNR_fwd_pred_ku(:,c) = rtadalinemex(CNR_fwd_ku(:,c),3,4,1e-6,[0.3 0.3
0.4],CNR_fwd_ku(1,c));
CNR_fwd_pred_V(:,c) = rtadalinemex(CNR_fwd_V(:,c),3,4,1e-6,[0.3 0.3
0.4],CNR_fwd_V(1,c));
CNR_fwd_pred_ku(1:86400-4,c) = CNR_fwd_pred_ku(5:86400,c);
CNR_fwd_pred_ku(86400-3:86400,c) = CNR_fwd_pred_ku(86400-4,c);
CNR_fwd_pred_V(1:86400-4,c) = CNR_fwd_pred_V(5:86400,c);
CNR_fwd_pred_V(86400-3:86400,c) = CNR_fwd_pred_V(86400-4,c);
end
disp('Done C/N calculations');

tic;
for csec = 1:(day-1)*86400+start_time-1
    for c = 1:Nstations_tot
        if csec == inter_session_start(call_count(c),c)
            active_station_info(find(active_station_info==0,1)) = c;
        end
        if csec == inter_session_end(call_count(c),c)
            active_station_info(find(active_station_info==c,1)) = 0;
            if call_count(c) < len
                call_count(c) = call_count(c) + 1;
            end
        end
    end
end
disp(['Time to create active_station_info: ' num2str(toc)]);
disp('Done initialisation');
for csec = start_time:end_time

```

```

if csec==1
    esec = 1;
else
    esec = csec - 1;
end
if csec < 4
    fsec = 4;
else
    fsec = csec;
end
mult = (day-1)*86400;
tic;
% Assign active station with CNR from one of the active channels
for c = 1:Nstations_tot
    if mult+csec == inter_session_start(call_count(c),c)
        active_station_info(find(active_station_info==0,1)) = c;
    end
    if mult+csec == inter_session_end(call_count(c),c)
        active_station_info(find(active_station_info==c,1)) = 0;
        if call_count(c) < len
            call_count(c) = call_count(c) + 1;
        end
    end
end
active_sum = sum(active_station_info & ones(Nstations,1));
active_index = find(active_station_info~=0);
if active_sum

    ind_ku = linspace(1,no_bursts_per_channel_ku,no_bursts_per_channel_ku);
    val_ku = 10000*ones(no_bursts_per_channel_ku,1);
    outage_vector = zeros(active_sum,1);
    for c = 1:active_sum
        mc = 1;
        notfilled = 1;
        if any(CNR_fwd_pred_ku(fsec-3:fsec,active_index(c)) > CNR_req_fwd_ku)
            while notfilled && mc <= no_bursts_per_channel_ku
                if mean(CNR_fwd_pred_V(fsec-3:fsec,active_index(c))) < val_ku(mc)
                    val_ku(mc+1:no_bursts_per_channel_ku)=val_ku(mc:no_bursts_per_channel_ku-1);
                    val_ku(mc) = mean(CNR_fwd_pred_ku(fsec-
3:fsec,active_index(c)));
                    ind_ku(mc+1:no_bursts_per_channel_ku)=ind_ku(mc:no_bursts_per_channel_ku-1);
                    ind_ku(mc) = c;
                    notfilled = 0;
                end
                mc = mc + 1;
            end
        end
    end
end

% CONTROL CHANNELS
% Generate protocol frames
data_fwd_ctrl_nccflc = randint(848,1); % random placeholder for NCC FLC
data_fwd_ctrl_eack = randint(96,1); % random placeholder for EACK
data_rtn_ctrl_rcst = randint(20,1); % random placeholder for RCST
data_rtn_ctrl_ecs = randint(60,1); % random placeholder for ECS
% Encode
data_coded_fwd_ctrl_nccflc = turbocodeDVB(data_fwd_ctrl_nccflc,424,'1/2'); %
turbo encode data
data_coded_fwd_ctrl_eack = turbocodeDVB(data_fwd_ctrl_eack,48,'1/2'); % turbo
encode data
data_coded_rtn_ctrl_rcst = convenc(data_rtn_ctrl_rcst,conv_trel); %
convolutionally encode data
data_coded_rtn_ctrl_ecs = convenc(data_rtn_ctrl_ecs,conv_trel); %
convolutionally encode data
% Convert to integer format if necessary
data_coded_int_rtn_ctrl_rcst = bit2intmex(data_coded_rtn_ctrl_rcst,2); %
convert 2 bits to integer
data_coded_int_rtn_ctrl_ecs = bit2intmex(data_coded_rtn_ctrl_ecs,2); % convert
2 bits to integer
% Modulate
data_modulated_fwd_ctrl_nccflc =
genqammod(data_coded_fwd_ctrl_nccflc,const_BPSK); % Modulate with BPSK
data_modulated_fwd_ctrl_eack = genqammod(data_coded_fwd_ctrl_eack,const_BPSK);
% Modulate with BPSK

```

```

data_modulated_rtn_ctrl_rcst(:,1:8) =
genqammod(data_coded_int_rtn_ctrl_rcst*ones(1,8),const_QPSK); % Modulate with QPSK
data_modulated_rtn_ctrl_ecs =
genqammod(data_coded_int_rtn_ctrl_rcst, const_QPSK); % Modulate with QPSK
for cfrm = 1:50 % for 50 frames per second
    % Add noise to single link
    data_noisy_up_fwd_ctrl_nccflc =
awgn(data_modulated_fwd_ctrl_nccflc,CNR_fwd_up_ctrl(csec),'measured');
    data_noisy_up_fwd_ctrl_eack =
awgn(data_modulated_fwd_ctrl_eack,CNR_fwd_up_ctrl(csec),'measured');

    for c = 1:active_sum % for the number of stations that are active
        CNR_fwd_ctrl_lin = 1/(1/(10^(CNR_fwd_up_ctrl(csec)/10)) +
1/(10^(CNR_fwd_down_ctrl(csec,active_index(c))/10)));
        CNR_rtn_ctrl_lin =
1/(1/(10^(CNR_rtn_up_ctrl(csec,active_index(c))/10)) +
1/(10^(CNR_rtn_down_ctrl(csec)/10)));
        % Add noise to multiple links
        data_noisy_down_fwd_ctrl_nccflc =
awgn(data_noisy_up_fwd_ctrl_nccflc,CNR_fwd_down_ctrl(csec,active_index(c)),'measured');
        data_noisy_down_fwd_ctrl_eack =
awgn(data_noisy_up_fwd_ctrl_eack,CNR_fwd_down_ctrl(csec,active_index(c)),'measured');
        data_noisy_up_rtn_ctrl_rcst =
awgn(data_modulated_rtn_ctrl_rcst,CNR_rtn_up_ctrl(csec,active_index(c)),'measured');
        data_noisy_up_rtn_ctrl_ecs =
awgn(data_modulated_rtn_ctrl_ecs,CNR_rtn_up_ctrl(csec,active_index(c)),'measured');
        data_noisy_down_rtn_ctrl_rcst =
awgn(data_noisy_up_rtn_ctrl_rcst,CNR_rtn_down_ctrl(csec),'measured');
        data_noisy_down_rtn_ctrl_ecs =
awgn(data_noisy_up_rtn_ctrl_ecs,CNR_rtn_down_ctrl(csec),'measured');
        % Convert symbols to log likelihood ratios
        LLR2_nccflc =
mod2llr(data_noisy_down_fwd_ctrl_nccflc,const_BPSK,CNR_fwd_ctrl_lin,2);
        LLR2_eack =
mod2llr(data_noisy_down_fwd_ctrl_eack,const_BPSK,CNR_fwd_ctrl_lin,2);
        for c2 = 1:8
            LLR2_rcst(:,c2) =
mod2llr(data_noisy_down_rtn_ctrl_rcst(:,c2),const_QPSK,1,4);
        end
        LLR2_ecs = mod2llr(data_noisy_down_rtn_ctrl_ecs,const_QPSK,1,4);
        % Covert to bitwise log likelihood ratios
        LLRs_nccflc = Somap(LLR2_nccflc,0);
        LLRs_eack = Somap(LLR2_eack,0);
        for c2 = 1:8
            LLRs_rcst(:,c2) = Somap(LLR2_rcst(:,c2),0);
        end
        LLRs_ecs = Somap(LLR2_ecs,0);
        % Decode
        data_decoded_fwd_ctrl_nccflc =
turboencodeDVB(LLRs_nccflc,424,'1/2',8,0,data_fwd_ctrl_nccflc);
        data_decoded_fwd_ctrl_eack =
turboencodeDVB(LLRs_eack,48,'1/2',8,0,data_fwd_ctrl_eack);
        for c2 = 1:8
            [x,qcode] = quantiz(LLRs_rcst(:,c2),[-3 -2 -1 0 1 2 3]',[0 1 2 3 4
5 6 7]'); % Values in qcode are between 0 and 2^3-1.
            data_decoded_rtn_ctrl_rcst(:,c2) =
vitdec(qcode,conv_trel,20,'trunc','soft',3);
        end
        [x,qcode] = quantiz(LLRs_ecs,[-3 -2 -1 0 1 2 3]',[0 1 2 3 4 5 6 7]');
        % Values in qcode are between 0 and 2^3-1.
        data_decoded_rtn_ctrl_ecs =
vitdec(qcode,conv_trel,45,'trunc','soft',3);
        % Count errors
        errors_nccflc =
sum(xor(fix((sign(data_decoded_fwd_ctrl_nccflc)+1)/2),data_fwd_ctrl_nccflc));
        errors_eack =
sum(xor(fix((sign(data_decoded_fwd_ctrl_eack)+1)/2),data_fwd_ctrl_eack));
        for c2 = 1:8
            errors_rcst_ind(c2) =
sum(xor(data_decoded_rtn_ctrl_rcst(:,c2),data_rtn_ctrl_rcst));
        end
        errors_rcst = sum(errors_rcst_ind);
        errors_ecs = sum(xor(data_decoded_rtn_ctrl_ecs,data_rtn_ctrl_ecs));
        station_no = active_station_info(active_index(c));
        biterrors_fwd_ctrl_nccflc(station_no) =
biterrors_fwd_ctrl_nccflc(station_no) + errors_nccflc;

```

```

        biterrors_fwd_ctrl_eack(station_no) =
biterrors_fwd_ctrl_eack(station_no) + errors_eack;
        biterrors_rtn_ctrl_rcst(station_no) =
biterrors_rtn_ctrl_rcst(station_no) + errors_rcst;
        biterrors_rtn_ctrl_ecs(station_no) =
biterrors_rtn_ctrl_ecs(station_no) + errors_ecs;
        if errors_nccflc
            frameerrors_fwd_ctrl_nccflc(station_no) =
frameerrors_fwd_ctrl_nccflc(station_no) + 1;
        end
        if errors_eack
            frameerrors_fwd_ctrl_eack(station_no) =
frameerrors_fwd_ctrl_eack(station_no) + 1;
        end
        if errors_rcst
            frameerrors_rtn_ctrl_rcst(station_no) =
frameerrors_rtn_ctrl_rcst(station_no) + 1;
        end
        if errors_ecs
            frameerrors_rtn_ctrl_ecs(station_no) =
frameerrors_rtn_ctrl_ecs(station_no) + 1;
        end
        if station_no == spec_stat
            biterrors_fwd_ctrl_specific(csec) =
biterrors_fwd_ctrl_specific(csec) + errors_nccflc + errors_eack;
            biterrors_rtn_ctrl_specific(csec) =
biterrors_rtn_ctrl_specific(csec) + errors_rcst + errors_ecs;
        end
    end
end
% TRAFFIC CHANNEL
for c = 1:active_sum
    station_no = active_station_info(active_index(c)); %spec_stat;
    % Generate protocol frames
    data_fwd_traf = randint(424,1); % random placeholder for forward traffic
    data_rtn_traf = randint(424,1); % random placeholder for return traffic
    if any (ind_ku == c) % Do ku-band channel
        FMT_type(csec,active_index(c)) = 10;
        CNR_fwd_traf_lin = 1/(1/(10^(CNR_fwd_up_ku(csec)/10)) +
1/(10^(CNR_fwd_down_ku(csec,active_index(c))/10)));
        CNR_rtn_traf_lin = 1/(1/(10^(CNR_rtn_up_ku(csec,active_index(c))/10))
+ 1/(10^(CNR_rtn_down_ku(csec)/10)));
        % Encode
        data_coded_fwd_traf = turbocodeDVB(data_fwd_traf,212,'1/2'); % turbo
    encode data
        data_coded_rtn_traf = turbocodeDVB(data_rtn_traf,212,'1/2'); % turbo
    encode data
        % Modulate
        data_coded_int_fwd_traf = bit2intmex(data_coded_fwd_traf,2); % convert
2 bits to integer
        data_coded_int_rtn_traf = bit2intmex(data_coded_rtn_traf,2); % convert
2 bits to integer
        data_modulated_fwd_traf =
genqammod(data_coded_int_fwd_traf*ones(1,27),const_QPSK); % Modulate with QPSK
        data_modulated_rtn_traf =
genqammod(data_coded_int_rtn_traf*ones(1,4),const_QPSK); % Modulate with QPSK
        for cfrm = 1:50
            % Add noise to a number of ATM frames according to protocol
            % for each frame
            data_noisy_up_fwd_traf =
awgn(data_modulated_fwd_traf,CNR_fwd_up_ku(csec),'measured');
            data_noisy_up_rtn_traf =
awgn(data_modulated_rtn_traf,CNR_rtn_up_ku(csec,active_index(c)),'measured');
            data_noisy_down_fwd_traf =
awgn(data_noisy_up_fwd_traf,CNR_fwd_down_ku(csec,active_index(c)),'measured');
            data_noisy_down_rtn_traf =
awgn(data_noisy_up_rtn_traf,CNR_rtn_down_ku(csec),'measured');
            % Calculate log-likelihoods & decode
            for c2 = 1:27
                LLR2_fwd_traf =
mod2llr(data_noisy_down_fwd_traf(:,c2),const_QPSK,CNR_fwd_traf_lin,4);
                LLRs_fwd_traf = Somap(LLR2_fwd_traf,0);
                data_decoded_fwd_traf =
turbodecodeDVB(LLRs_fwd_traf,212,'1/2',8,0,data_fwd_traf);
                errors_fwd_traf =
sum(xor(fix((sign(data_decoded_fwd_traf)+1)/2),data_fwd_traf));
            end
        end
    end
end

```



```

        biterrors_fwd_traf(station_no) =
biterrors_fwd_traf(station_no) + errors_fwd_traf;
        if station_no == spec_stat
            biterrors_fwd_traf_specific(csec) =
biterrors_fwd_traf_specific(csec) + errors_fwd_traf;
        end
    end
    for c2 = 1:4
        LLR2_rtn_traf =
mod2llr(data_noisy_down_rtn_traf(:,c2),const_QPSK,CNR_rtn_traf_lin,4);
        LLRs_rtn_traf = Somap(LLR2_rtn_traf,0);
        data_decoded_rtn_traf =
turbodecodeDVB(LLRs_rtn_traf,212,'1/2',8,0,data_rtn_traf);
        errors_rtn_traf =
sum(xor(fix((sign(data_decoded_rtn_traf)+1)/2),data_rtn_traf));
        biterrors_rtn_traf(station_no) =
biterrors_rtn_traf(station_no) + errors_rtn_traf;
        if station_no == spec_stat
            biterrors_rtn_traf_specific(csec) =
biterrors_rtn_traf_specific(csec) + errors_rtn_traf;
        end
    end
end
else % Do V-band channel
    CNR_fwd_traf_lin = 1/(1/(10^(CNR_fwd_up_V(csec)/10)) +
1/(10^(CNR_fwd_down_V(csec,active_index(c))/10)));
    CNR_rtn_traf_lin = 1/(1/(10^(CNR_rtn_up_V(csec,active_index(c))/10)) +
1/(10^(CNR_rtn_down_V(csec)/10)));
    % Determine FMT scheme
    if all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(1)+scint_margin)
        code_rate = FMT_code_rates[97];
        mod_type = FMT_bits_per_symbol(1);
        FMT_type(csec,active_index(c)) = 1;

        elseif all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(2)+scint_margin)
        code_rate = FMT_code_rates[97];
        mod_type = FMT_bits_per_symbol(2);
        FMT_type(csec,active_index(c)) = 2;

        elseif all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(3)+scint_margin)
        code_rate = FMT_code_rates[97];
        mod_type = FMT_bits_per_symbol(3);
        FMT_type(csec,active_index(c)) = 3;

        elseif all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(4)+scint_margin)
        code_rate = FMT_code_rates[129];
        mod_type = FMT_bits_per_symbol(4);
        FMT_type(csec,active_index(c)) = 4;

        elseif all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(5)+scint_margin)
        code_rate = FMT_code_rates[6];
        mod_type = FMT_bits_per_symbol(5);
        FMT_type(csec,active_index(c)) = 5;

        elseif all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(6)+scint_margin)
        code_rate = FMT_code_rates[97];
        mod_type = FMT_bits_per_symbol(6);
        FMT_type(csec,active_index(c)) = 6;

        elseif all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(7)+scint_margin)
        code_rate = FMT_code_rates{7};
        mod_type = FMT_bits_per_symbol(7);
        FMT_type(csec,active_index(c)) = 7;

        elseif all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(8)+scint_margin)
        code_rate = FMT_code_rates{8};
        mod_type = FMT_bits_per_symbol(8);
        FMT_type(csec,active_index(c)) = 8;

```

```

elseif all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(9)+scint_margin)
    code_rate = FMT_code_rates{9};
    mod_type = FMT_bits_per_symbol(9);
    FMT_type(csec,active_index(c)) = 9;

else
    if FMT_type(csec-1,active_index(c)) < 10
        code_rate = FMT_code_rates{FMT_type(csec-1,active_index(c))};
        mod_type = FMT_bits_per_symbol(FMT_type(csec-
1,active_index(c)));
        FMT_type(csec,active_index(c)) = FMT_type(csec-
1,active_index(c));
    elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(1)+scint_margin
        code_rate = FMT_code_rates{1};
        mod_type = FMT_bits_per_symbol(1);
        FMT_type(csec,active_index(c)) = 1;

        elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(2)+scint_margin
        code_rate = FMT_code_rates{2};
        mod_type = FMT_bits_per_symbol(2);
        FMT_type(csec,active_index(c)) = 2;

        elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(3)+scint_margin
        code_rate = FMT_code_rates{3};
        mod_type = FMT_bits_per_symbol(3);
        FMT_type(csec,active_index(c)) = 3;

        elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(4)+scint_margin
        code_rate = FMT_code_rates{4};
        mod_type = FMT_bits_per_symbol(4);
        FMT_type(csec,active_index(c)) = 4;

        elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(5)+scint_margin
        code_rate = FMT_code_rates{5};
        mod_type = FMT_bits_per_symbol(5);
        FMT_type(csec,active_index(c)) = 5;

        elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(6)+scint_margin
        code_rate = FMT_code_rates{6};
        mod_type = FMT_bits_per_symbol(6);
        FMT_type(csec,active_index(c)) = 6;

        elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(7)+scint_margin
        code_rate = FMT_code_rates{7};
        mod_type = FMT_bits_per_symbol(7);
        FMT_type(csec,active_index(c)) = 7;

        elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(8)+scint_margin
        code_rate = FMT_code_rates{8};
        mod_type = FMT_bits_per_symbol(8);
        FMT_type(csec,active_index(c)) = 8;

        elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(9)+scint_margin
        code_rate = FMT_code_rates{9};
        mod_type = FMT_bits_per_symbol(9);
        FMT_type(csec,active_index(c)) = 9;

    end

end
% Encode
data_coded_fwd_traf = turbocodeDVB(data_fwd_traf,212,code_rate); %
turbo encode data
data_coded_rtn_traf = turbocodeDVB(data_rtn_traf,212,code_rate); %
turbo encode data
% Modulate
if mod_type == 2

```



```

        data_coded_int_fwd_traf = bit2intmex(data_coded_fwd_traf,2); %
convert 2 bits to integer
        data_coded_int_rtn_traf = bit2intmex(data_coded_rtn_traf,2); %
convert 2 bits to integer
        data_modulated_fwd_traf =
genqammod(data_coded_int_fwd_traf*ones(1,27),const_QPSK); % Modulate with QPSK
        data_modulated_rtn_traf =
genqammod(data_coded_int_rtn_traf*ones(1,4),const_QPSK); % Modulate with QPSK
    else
        data_modulated_fwd_traf =
genqammod(data_coded_fwd_traf*ones(1,27),const_BPSK); % Modulate with BPSK
        data_modulated_rtn_traf =
genqammod(data_coded_rtn_traf*ones(1,4),const_BPSK); % Modulate with BPSK
    end
    for cfrm = 1:50
        % Add noise to a number of ATM frames according to protocol
        % for each frame

        data_noisy_down_fwd_traf =
awgn(data_modulated_fwd_traf,CNR_fwd_V(csec,active_index(c)),'measured');
        data_noisy_down_rtn_traf =
awgn(data_modulated_rtn_traf,CNR_rtn_V(csec),'measured');
        % Calculate log-likelihoods & decode
        for c2 = 1:27
            if mod_type == 2
                LLR2_fwd_traf =
mod2llr(data_noisy_down_fwd_traf,const_QPSK,CNR_fwd_traf_lin,4);
            else
                LLR2_fwd_traf =
mod2llr(data_noisy_down_fwd_traf,const_BPSK,CNR_fwd_traf_lin,2);
            end
            LLRs_fwd_traf = Somap(LLR2_fwd_traf,0);
            data_decoded_fwd_traf =
turbodecodeDVB(LLRs_fwd_traf,212,code_rate,8,0,data_fwd_traf);
            errors_fwd_traf =
sum(xor(fix((sign(data_decoded_fwd_traf)+1)/2),data_fwd_traf));
            biterrors_fwd_traf(station_no) =
biterrors_fwd_traf(station_no) + errors_fwd_traf;
            if station_no == spec_stat
                biterrors_fwd_traf_specific(csec) =
biterrors_fwd_traf_specific(csec) + errors_fwd_traf;
            end
        end
        for c2 = 1:4
            if mod_type == 2
                LLR2_rtn_traf =
mod2llr(data_noisy_down_rtn_traf,const_QPSK,CNR_rtn_traf_lin,4);
            else
                LLR2_rtn_traf =
mod2llr(data_noisy_down_rtn_traf,const_BPSK,CNR_rtn_traf_lin,2);
            end
            LLRs_rtn_traf = Somap(LLR2_rtn_traf,0);
            data_decoded_rtn_traf =
turbodecodeDVB(LLRs_rtn_traf,212,code_rate,8,0,data_rtn_traf);
            errors_rtn_traf =
sum(xor(fix((sign(data_decoded_rtn_traf)+1)/2),data_rtn_traf));
            biterrors_rtn_traf(station_no) =
biterrors_rtn_traf(station_no) + errors_rtn_traf;
            if station_no == spec_stat
                biterrors_rtn_traf_specific(csec) =
biterrors_rtn_traf_specific(csec) + errors_rtn_traf;
            end
        end
    end
end
end
end
disp(['Time elapsed: ' num2str(toc)]);
end
BER_fwd_traf_specific = biterrors_fwd_traf_specific/(50*27*424);
BER_rtn_traf_specific = biterrors_rtn_traf_specific/(50*4*424);
BER_fwd_ctrl_specific = biterrors_fwd_ctrl_specific/(50*(848+96));
BER_rtn_ctrl_specific = biterrors_rtn_ctrl_specific/(50*(8*20+60));
save('Final_detailed_test_results_4station');

```

Appendix V2

Final event analysis with 1 ku band station

```
clear;
load('daysInfo2');
load('inter_session_indices_620');
load('SystemDefinitions');
load('Fwd_ctrl_info','Rs_fwd_ctrl','B_fwd_ctrl');
load('Rtn_ctrl_info','Rs_rtn_ctrl','B_rtn_ctrl');
load('Fwd_traf_info2','Rs_fwd_traf','Rs_fwd_traf_ku','B','B_ku');
load('Rtn_traf_info2','Rs_rtn_traf','Rs_rtn_traf_ku','B_per_channel','B_per_channel_ku');
load('OutboundTraffic_mixed2','CtoN_req','CtoN_req_ku','CtoN_tot','CtoN_tot_ku','durout','marginout');
CNR_req_fwd = CtoN_req;
CNR_req_fwd_ku = CtoN_req_ku;
CNR_ca_fwd = CtoN_tot;
CNR_ca_fwd_ku = CtoN_tot_ku;
duration_fwd = durout;
margins_fwd = marginout;
load('InboundTraffic_mixed2','CtoN_req','CtoN_req_ku','CtoN_tot','CtoN_tot_ku','durin','marginin');
CNR_req_rtn = CtoN_req;
CNR_req_rtn_ku = CtoN_req_ku;
CNR_ca_rtn = CtoN_tot;
CNR_ca_rtn_ku = CtoN_tot_ku;
duration_rtn = durin;
margins_rtn = marginin;
clear CtoN_req CtoN_req_ku CtoN_tot CtoN_tot_ku durin durout marginin marginout;
load('Final_detailed_find_events_results_620_2','event_list');

event_no = 92;
day = event_list(event_no).day;
start_time = event_list(event_no).start;
end_time = event_list(event_no).end;
spec_stat = event_list(event_no).station;
spec_series = event_list(event_no).series;

% System info
Nstations = find(activity_cdf==0,1)-1;
Phub_traf_V = 5;
Phub_traf_ku = 199;
Phub_ctrl = 1;
Pvsat_traf_V = 0.8;
Pvsat_traf_ku = 2*0.8;
Pvsat_ctrl = 2*0.2;
TransponderBW = 36e6;
Psat_fwd_traf = Psat*B/TransponderBW;
Psat_rtn_traf = Psat*B_per_channel/TransponderBW;
Psat_fwd_traf_ku = Psat*B_ku/TransponderBW;
Psat_rtn_traf_ku = Psat*B_per_channel_ku/TransponderBW;
Psat_fwd_ctrl = Psat*B_fwd_ctrl/TransponderBW;
Psat_rtn_ctrl = Psat*B_rtn_ctrl/TransponderBW;
Tsysca_sat_V = 900;
Tsysca_sat_ku = 500;
Tsysca_ground_V = 350;
Tsysca_ground_ku = 150;
Leoc_hub = 1;
Leoc_vsat = 3;
Lmisc_hub = 1;
Lmisc_vsat = 1;
FMT_CNR_reqs = [8.5 7.9 7.2 6.7 5.8 4.9 4.2 3.7 2.8];
FMT_code_rates = {'6/7','4/5','3/4','2/3','1/2','4/5','3/4','2/3','1/2'};
FMT_bits_per_symbol = [2 2 2 2 1 1 1 1];
no_bursts_per_channel_ku = 1;
scint_margin = 1;

% design lpf & hpf
n = 40;
fs = 1;
```

```

cutoff = 20e-3;
Wn = cutoff/(fs/2);
blpf = fir1(n,Wn);
a = 1;

% Initialize working memory
ts40 = zeros(86400,1);
ts50 = zeros(86400,1);
gas40 = zeros(86400,1);
gas50 = zeros(86400,1);
CNR_fwd_down_V = zeros(86400,Nstations);
CNR_rtn_up_V = zeros(86400,Nstations);
CNR_fwd_down_ku = zeros(86400,Nstations);
CNR_rtn_up_ku = zeros(86400,Nstations);
CNR_fwd_down_ctrl = zeros(86400,Nstations);
CNR_rtn_up_ctrl = zeros(86400,Nstations);
CNR_fwd_V = zeros(86400,Nstations);
CNR_rtn_V = zeros(86400,Nstations);
CNR_fwd_pred_ku = zeros(86400,Nstations);
active_station_info = zeros(Nstations,1);
biterrors_fwd_ctrl_nccflc = zeros(Nstations_tot,1);
biterrors_fwd_ctrl_eack = zeros(Nstations_tot,1);
biterrors_rtn_ctrl_rcst = zeros(Nstations_tot,1);
biterrors_rtn_ctrl_ecs = zeros(Nstations_tot,1);
frameerrors_fwd_ctrl_nccflc = zeros(Nstations_tot,1);
frameerrors_fwd_ctrl_eack = zeros(Nstations_tot,1);
frameerrors_rtn_ctrl_rcst = zeros(Nstations_tot,1);
frameerrors_rtn_ctrl_ecs = zeros(Nstations_tot,1);
biterrors_fwd_traf = zeros(Nstations_tot,1);
biterrors_rtn_traf = zeros(Nstations_tot,1);
frameerrors_fwd_traf = zeros(Nstations_tot,1);
frameerrors_rtn_traf = zeros(Nstations_tot,1);
biterrors_fwd_traf_specific = zeros(86400,1);
biterrors_rtn_traf_specific = zeros(86400,1);
biterrors_fwd_ctrl_specific = zeros(86400,1);
biterrors_rtn_ctrl_specific = zeros(86400,1);
FMT_type = ones(86400,58);
errors_rcst_ind = zeros(8,1);
call_count = ones(Nstations_tot,1);
Nconc = zeros(Nstations,1);
[len,dum] = size(inter_session_start);
const_BPSK = [1 -1];
const_QPSK = [1;j;-j;-1];
pwr_BPSK = sum(abs(const_BPSK).^2)/length(const_BPSK);
pwr_QPSK = sum(abs(const_QPSK).^2)/length(const_QPSK);
conv_trel = poly2trellis(7,[171,133]);

% Do hub side CNR calculations
date = daysInfo(day).date;
dummycell = load(['Data3/S40_' date '.mat']);
eval(['ts40 = dummycell.S40_' date '(:,2);']);
eval(['gas40 = dummycell.S40_' date '(:,2) - dummycell.S40_' date '(:,1);']);
dummycell = load(['Data3/S50_' date '.mat']);
eval(['ts50 = dummycell.S50_' date '(:,2);']);
eval(['gas50 = dummycell.S50_' date '(:,2) - dummycell.S50_' date '(:,1);']);
% Apply timeshift for site diversity
timeshift = 14.5*60;
for c = 1:86400-timeshift
    ts40(c,1) = min(ts40(c,1),ts40(c+timeshift,1));
    ts50(c,1) = min(ts50(c,1),ts50(c+timeshift,1));
end
[rain40,zr] = filter(blpf,a,ts40,ts40(1)*ones(n,1));
rain40(1:86400-n/2) = rain40(n/2+1:86400);
rain40(1:n/2) = ts40(1:n/2);
rain40(86400-n/2+1:86400) = ts40(86400-n/2+1:86400);
nogas40 = rain40 - gas40;
scint40 = ts40 - rain40;
gas14 = 0.7;
gas12 = 0.5;
ts14 = fscale(39.5,14,nogas40) + sscale(39.5,14,scint40,0.6,0.6,30) + gas14;
ts12 = fscale(39.5,12,nogas40) + sscale(39.5,12,scint40,0.6,0.6,30) + gas12;
CNR_fwd_up_V =
att2cnr(ts50,gas50,Gtx_hub_V+10*log10(Phub_traf_V),Grx_satellite_V,Tsysca_sat_V,ulink_
freq,Rs_fwd_traf,Leoc_hub,Lmisc_hub,0);

```

```

CNR_rtn_down_V =
att2cnr(tsl40,gas40,Gtx_satellite_V+10*log10(Psat_rtn_traf),Grx_hub_V,Tsysca_ground_V,d
link_freq,Rs_rtn_traf,Leoc_hub,Lmisc_hub,1);
CNR_fwd_up_ku =
att2cnr(tsl14,gas14*ones(86400,1),Gtx_hub_ku+10*log10(Phub_traf_ku),Grx_satellite_ku,Ts
ysca_sat_ku,ulink_freq_ku,Rs_fwd_traf_ku,Leoc_hub,Lmisc_hub,0);
CNR_rtn_down_ku =
att2cnr(tsl2,gas12*ones(86400,1),Gtx_satellite_ku+10*log10(Psat_rtn_traf_ku),Grx_hub_k
u,Tsysca_ground_ku,dlink_freq_ku,Rs_rtn_traf_ku,Leoc_hub,Lmisc_hub,1);
CNR_fwd_up_ctrl =
att2cnr(tsl14,gas14*ones(86400,1),Gtx_hub_ku+10*log10(Phub_ctrl),Grx_satellite_ku,Tsysc
a_sat_ku,ulink_freq_ku,Rs_fwd_ctrl,Leoc_hub,Lmisc_hub,0);
CNR_rtn_down_ctrl =
att2cnr(tsl2,gas12*ones(86400,1),Gtx_satellite_ku+10*log10(Psat_rtn_ctrl),Grx_hub_ku,T
sysca_ground_ku,dlink_freq_ku,Rs_rtn_ctrl,Leoc_hub,Lmisc_hub,1);
CNR_fwd_up_ku0 =
att2cnr(gas14,gas14,Gtx_hub_ku+10*log10(Phub_traf_ku),Grx_satellite_ku,Tsysca_sat_ku,u
link_freq_ku,Rs_fwd_traf_ku,Leoc_hub,Lmisc_hub,0);
CNR_fwd_up_V0 =
att2cnr(gas50(:,1),gas50(:,1),Gtx_hub_V+10*log10(Phub_traf_V),Grx_satellite_V,Tsysca_s
at_V,ulink_freq,Rs_fwd_traf,Leoc_hub,Lmisc_hub,0);

% Create lookup table for dNrain, indexed by attenuation
dNrain_ku = zeros(101,1);
CNR_reduction_ku = zeros(101,1);
CNR_reduction_hub_ku = zeros(101,1);
Taca_ku = 275*(1-10^(-(gas12/10)));
for att = 0:100
    Tarain_ku = 275*(1-10^(-(gas12+att)/10));
    dTsys_ku = Tarain_ku - Taca_ku;
    Tsyrain_ku = Tsysca_ground_ku + dTsys_ku;
    dNrain_ku(att+1) = 10*log10(Tsyrain_ku/Tsysca_ground_ku);
    CNR_reduction_ku(att+1) = att + dNrain_ku(att+1);
    CNR_reduction_hub_ku(att+1) = fscale(12,14,att) + att + dNrain_ku(att+1);
end
Taca_V = 275*(1-10^(-(1/10)));

% Calculate forward downlink for the hub.
CNR_fwd_down_hub_ku =
att2cnr(tsl2(:,1),gas12*ones(86400,1),Gtx_satellite_ku+10*log10(Psat_fwd_traf_ku),Grx_
hub_ku,Tsysca_ground_ku,dlink_freq_ku,Rs_fwd_traf_ku,Leoc_hub,Lmisc_hub,1);
CNR_fwd_down_hub_ku0 =
att2cnr(gas12,gas12,Gtx_satellite_ku+10*log10(Psat_fwd_traf_ku),Grx_hub_ku,Tsysca_grou
nd_ku,dlink_freq_ku,Rs_fwd_traf_ku,Leoc_hub,Lmisc_hub,1);
CNR_fwd_hub_0 = 10*log10(1./((1./(10.^(CNR_fwd_up_ku0/10))) +
(1./10.^(CNR_fwd_down_hub_ku0/10))));
CNR_fwd_hub_ku = 10*log10(1./((1./(10.^(CNR_fwd_up_ku/10))) +
(1./10.^(CNR_fwd_down_hub_ku/10))));
CNR_fwd_pred_hub_ku = rtadalinemex(CNR_fwd_hub_ku,3,1,1e-6,[0.3 0.3
0.4],CNR_fwd_hub_ku(1));

AttNoise_ku = CNR_fwd_hub_0 - CNR_fwd_pred_hub_ku;
Att_ku = interp1(CNR_reduction_hub_ku,(0:100),AttNoise_ku,'linear','extrap');
Att_V = fscale(12,50,Att_ku);

disp('Done hub side calculations');

% Adjust overall CNR based on power control
CNR_fwd_up_V_pctrl = CNR_fwd_up_V;
for c = 1:86399
    if Att_V(c) > 6
        CNR_fwd_up_V_pctrl(c+1) = CNR_fwd_up_V(c+1) + 6;
    elseif Att_V(c) > 3
        CNR_fwd_up_V_pctrl(c+1) = CNR_fwd_up_V(c+1) + 3;
    end
end

% Do vsat side CNR calculations & total CNR calculations with power control
for c = 1:Nstations
    varday = rem(day+c,length(daysInfo));
    if varday == 0
        varday = length(daysInfo);
    end
    date = daysInfo(varday).date;
    dummycell = load(['Data3/S40_' date '.mat']);
    eval(['ts40 = dummycell.S40_' date '(:,2);']);

```

```

eval(['gas40 = dummycell.S40_' date '(:,2) - dummycell.S40_' date '(:,1)']);
dummycell = load(['Data3/S50_' date '.mat']);
eval(['ts50 = dummycell.S50_' date '(:,2)']);
eval(['gas50 = dummycell.S50_' date '(:,2) - dummycell.S50_' date '(:,1)']);
[rain40,zr] = filter(blpf,a,ts40,ts40(1)*ones(n,1));
rain40(1:86400-n/2) = rain40(n/2+1:86400);
rain40(86400-n/2+1:86400) = zr(1:n/2);
nogas40 = rain40 - gas40;
scint40 = ts40 - rain40;
ts14 = fscale(39.5,14,nogas40) + sscale(39.5,14,scint40,0.6,0.6,30) + gas14;
ts12 = fscale(39.5,12,nogas40) + sscale(39.5,12,scint40,0.6,0.6,30) + gas12;
CNR_fwd_down_V(:,c) =
att2cnr(ts40,gas40,Gtx_satellite_V+10*log10(Psat_fwd_traf),Grx_vsats_V,Tsysca_ground_V,
dlink_freq,rs_fwd_traf,Leoc_vsats,Lmisc_vsats,1);
CNR_rtn_up_V(:,c) =
att2cnr(ts50,gas50,Gtx_vsats_V+10*log10(Pvsats_traf_V),Grx_satellite_V,Tsysca_sats_V,ulink
k_freq,rs_rtn_traf,Leoc_vsats,Lmisc_vsats,0);
CNR_fwd_down_ku(:,c) =
att2cnr(ts12,gas12*ones(86400,1),Gtx_satellite_ku+10*log10(Psat_fwd_traf_ku),Grx_vsats_
ku,Tsysca_ground_ku,dlink_freq_ku,rs_fwd_traf_ku,Leoc_vsats,Lmisc_vsats,1);
CNR_rtn_up_ku(:,c) =
att2cnr(ts14,gas14*ones(86400,1),Gtx_vsats_ku+10*log10(Pvsats_traf_ku),Grx_satellite_ku,
Tsysca_sats_ku,ulink_freq_ku,rs_rtn_traf_ku,Leoc_vsats,Lmisc_vsats,0);
CNR_fwd_down_ctrl(:,c) =
att2cnr(ts12,gas12*ones(86400,1),Gtx_satellite_ku+10*log10(Psat_fwd_ctrl),Grx_vsats_ku,
Tsysca_ground_ku,dlink_freq_ku,rs_fwd_ctrl,Leoc_vsats,Lmisc_vsats,1);
CNR_rtn_up_ctrl(:,c) =
att2cnr(ts14,gas14*ones(86400,1),Gtx_vsats_ku+10*log10(Pvsats_ctrl),Grx_satellite_ku,Tsy
sca_sats_ku,ulink_freq_ku,rs_rtn_ctrl,Leoc_vsats,Lmisc_vsats,0);
CNR_fwd_down_ku0(:,c) =
att2cnr(gas12,gas12,Gtx_satellite_ku+10*log10(Psat_fwd_traf_ku),Grx_vsats_ku,Tsysca_gro
und_ku,dlink_freq_ku,rs_fwd_traf_ku,Leoc_vsats,Lmisc_vsats,1);
CNR_fwd_down_V0(:,c) =
att2cnr(gas40,gas40,Gtx_satellite_V+10*log10(Psat_fwd_traf),Grx_vsats_V,Tsysca_ground_V
,dlink_freq,rs_fwd_traf,Leoc_vsats,Lmisc_vsats,1);

CNR_fwd_V(:,c) = 10*log10(1./((1./(10.^(CNR_fwd_up_V_pctrl/10))) +
(1./10.^(CNR_fwd_down_V(:,c)/10))));
CNR_rtn_V(:,c) = 10*log10(1./((1./(10.^(CNR_rtn_up_V(:,c)/10))) +
(1./10.^(CNR_rtn_down_V/10))));
CNR_fwd_ku(:,c) = 10*log10(1./((1./(10.^(CNR_fwd_up_ku/10))) +
(1./10.^(CNR_fwd_down_ku(:,c)/10))));
CNR_rtn_ku(:,c) = 10*log10(1./((1./(10.^(CNR_rtn_up_ku(:,c)/10))) +
(1./10.^(CNR_rtn_down_ku/10))));
CNR_fwd_ku0(:,c) = 10*log10(1./((1./(10.^(CNR_fwd_up_ku0/10))) +
(1./10.^(CNR_fwd_down_ku0(:,c)/10))));
CNR_fwd_V0(:,c) = 10*log10(1./((1./(10.^(CNR_fwd_up_V0/10))) +
(1./10.^(CNR_fwd_down_V0(:,c)/10))));

CNR_fwd_pred_ku(:,c) = rtadalinemex(CNR_fwd_ku(:,c),3,4,1e-6,[0.3 0.3
0.4],CNR_fwd_ku(1,c));
CNR_fwd_pred_V(:,c) = rtadalinemex(CNR_fwd_V(:,c),3,4,1e-6,[0.3 0.3
0.4],CNR_fwd_V(1,c));
CNR_fwd_pred_ku(1:86400-4,c) = CNR_fwd_pred_ku(5:86400,c);
CNR_fwd_pred_ku(86400-3:86400,c) = CNR_fwd_pred_ku(86400-4,c);
CNR_fwd_pred_V(1:86400-4,c) = CNR_fwd_pred_V(5:86400,c);
CNR_fwd_pred_V(86400-3:86400,c) = CNR_fwd_pred_V(86400-4,c);
end
disp('Done C/N calculations');

tic;
for csec = 1:(day-1)*86400+start_time-1
    for c = 1:Nstations_tot
        if csec == inter_session_start(call_count(c),c)
            active_station_info(find(active_station_info==0,1)) = c;
        end
        if csec == inter_session_end(call_count(c),c)
            active_station_info(find(active_station_info==c,1)) = 0;
            if call_count(c) < len
                call_count(c) = call_count(c) + 1;
            end
        end
    end
end
disp(['Time to create active station info: ' num2str(toc)]);
disp('Done initialisation');
for csec = start_time:end_time

```

```

if csec==1
    esec = 1;
else
    esec = csec - 1;
end
if csec < 4
    fsec = 4;
else
    fsec = csec;
end
mult = (day-1)*86400;
tic;
% Assign active station with CNR from one of the active channels
for c = 1:Nstations_tot
    if mult+csec == inter_session_start(call_count(c),c)
        active_station_info(find(active_station_info==0,1)) = c;
    end
    if mult+csec == inter_session_end(call_count(c),c)
        active_station_info(find(active_station_info==c,1)) = 0;
        if call_count(c) < len
            call_count(c) = call_count(c) + 1;
        end
    end
end
active_sum = sum(active_station_info & ones(Nstations,1));
active_index = find(active_station_info~=0);
if active_sum

    ind_ku = linspace(1,no_bursts_per_channel_ku,no_bursts_per_channel_ku);
    val_ku = 10000*ones(no_bursts_per_channel_ku,1);
    outage_vector = zeros(active_sum,1);
    for c = 1:active_sum
        mc = 1;
        notfilled = 1;
        if any(CNR_fwd_pred_ku(fsec-3:fsec,active_index(c)) > CNR_req_fwd_ku)
            while notfilled && mc <= no_bursts_per_channel_ku
                if mean(CNR_fwd_pred_V(fsec-3:fsec,active_index(c))) < val_ku(mc)
                    val_ku(mc+1:no_bursts_per_channel_ku)=val_ku(mc:no_bursts_per_channel_ku-1);
                    val_ku(mc) = mean(CNR_fwd_pred_ku(fsec-
3:fsec,active_index(c)));
                    ind_ku(mc+1:no_bursts_per_channel_ku)=ind_ku(mc:no_bursts_per_channel_ku-1);
                    ind_ku(mc) = c;
                    notfilled = 0;
                end
                mc = mc + 1;
            end
        end
    end
end

% CONTROL CHANNELS
% Generate protocol frames
data_fwd_ctrl_nccflc = randint(848,1); % random placeholder for NCC FLC
data_fwd_ctrl_eack = randint(96,1); % random placeholder for EACK
data_rtn_ctrl_rcst = randint(20,1); % random placeholder for RCST
data_rtn_ctrl_ecs = randint(60,1); % random placeholder for ECS
% Encode
data_coded_fwd_ctrl_nccflc = turbocodeDVB(data_fwd_ctrl_nccflc,424,'1/2'); %
turbo encode data
data_coded_fwd_ctrl_eack = turbocodeDVB(data_fwd_ctrl_eack,48,'1/2'); % turbo
encode data
data_coded_rtn_ctrl_rcst = convenc(data_rtn_ctrl_rcst,conv_trel); %
convolutionally encode data
data_coded_rtn_ctrl_ecs = convenc(data_rtn_ctrl_ecs,conv_trel); %
convolutionally encode data
% Convert to integer format if necessary
data_coded_int_rtn_ctrl_rcst = bit2intmex(data_coded_rtn_ctrl_rcst,2); %
convert 2 bits to integer
data_coded_int_rtn_ctrl_ecs = bit2intmex(data_coded_rtn_ctrl_ecs,2); % convert
2 bits to integer
% Modulate
data_modulated_fwd_ctrl_nccflc =
genqammod(data_coded_fwd_ctrl_nccflc,const_BPSK); % Modulate with BPSK
data_modulated_fwd_ctrl_eack = genqammod(data_coded_fwd_ctrl_eack,const_BPSK);
% Modulate with BPSK

```



```

data_modulated_rtn_ctrl_rcst(:,1:8) =
genqammod(data_coded_int_rtn_ctrl_rcst*ones(1,8),const_QPSK); % Modulate with QPSK
data_modulated_rtn_ctrl_ecs =
genqammod(data_coded_int_rtn_ctrl_rcst, const_QPSK); % Modulate with QPSK
for cfrm = 1:50 % for 50 frames per second
    % Add noise to single link
    data_noisy_up_fwd_ctrl_nccflc =
awgn(data_modulated_fwd_ctrl_nccflc,CNR_fwd_up_ctrl(csec),'measured');
    data_noisy_up_fwd_ctrl_eack =
awgn(data_modulated_fwd_ctrl_eack,CNR_fwd_up_ctrl(csec),'measured');

    for c = 1:active_sum % for the number of stations that are active
        CNR_fwd_ctrl_lin = 1/(1/(10^(CNR_fwd_up_ctrl(csec)/10)) +
1/(10^(CNR_fwd_down_ctrl(csec,active_index(c))/10)));
        CNR_rtn_ctrl_lin =
1/(1/(10^(CNR_rtn_up_ctrl(csec,active_index(c))/10)) +
1/(10^(CNR_rtn_down_ctrl(csec)/10)));
        % Add noise to multiple links
        data_noisy_down_fwd_ctrl_nccflc =
awgn(data_noisy_up_fwd_ctrl_nccflc,CNR_fwd_down_ctrl(csec,active_index(c)),'measured')
;
        data_noisy_down_fwd_ctrl_eack =
awgn(data_noisy_up_fwd_ctrl_eack,CNR_fwd_down_ctrl(csec,active_index(c)),'measured');
        data_noisy_up_rtn_ctrl_rcst =
awgn(data_modulated_rtn_ctrl_rcst,CNR_rtn_up_ctrl(csec,active_index(c)),'measured');
        data_noisy_up_rtn_ctrl_ecs =
awgn(data_modulated_rtn_ctrl_ecs,CNR_rtn_up_ctrl(csec,active_index(c)),'measured');
        data_noisy_down_rtn_ctrl_rcst =
awgn(data_noisy_up_rtn_ctrl_rcst,CNR_rtn_down_ctrl(csec),'measured');
        data_noisy_down_rtn_ctrl_ecs =
awgn(data_noisy_up_rtn_ctrl_ecs,CNR_rtn_down_ctrl(csec),'measured');
        % Convert symbols to log likelihood ratios
        LLR2_nccflc =
mod2llr(data_noisy_down_fwd_ctrl_nccflc,const_BPSK,CNR_fwd_ctrl_lin,2);
        LLR2_eack =
mod2llr(data_noisy_down_fwd_ctrl_eack,const_BPSK,CNR_fwd_ctrl_lin,2);
        for c2 = 1:8
            LLR2_rcst(:,c2) =
mod2llr(data_noisy_down_rtn_ctrl_rcst(:,c2),const_QPSK,1,4);
        end
        LLR2_ecs = mod2llr(data_noisy_down_rtn_ctrl_ecs,const_QPSK,1,4);
        % Covert to bitwise log likelihood ratios
        LLRs_nccflc = Somap(LLR2_nccflc,0);
        LLRs_eack = Somap(LLR2_eack,0);
        for c2 = 1:8
            LLRs_rcst(:,c2) = Somap(LLR2_rcst(:,c2),0);
        end
        LLRs_ecs = Somap(LLR2_ecs,0);
        % Decode
        data_decoded_fwd_ctrl_nccflc =
turbo decodeDVB(LLRs_nccflc,424,'1/2',8,0,data_fwd_ctrl_nccflc);
        data_decoded_fwd_ctrl_eack =
turbo decodeDVB(LLRs_eack,48,'1/2',8,0,data_fwd_ctrl_eack);
        for c2 = 1:8
            [x,qcode] = quantiz(LLRs_rcst(:,c2),[-3 -2 -1 0 1 2 3]',[0 1 2 3 4
5 6 7]'); % Values in qcode are between 0 and 2^3-1.
            data_decoded_rtn_ctrl_rcst(:,c2) =
vitdec(qcode,conv_trel,20,'trunc','soft',3);
        end
        [x,qcode] = quantiz(LLRs_ecs,[-3 -2 -1 0 1 2 3]',[0 1 2 3 4 5 6 7]');
        % Values in qcode are between 0 and 2^3-1.
        data_decoded_rtn_ctrl_ecs =
vitdec(qcode,conv_trel,45,'trunc','soft',3);
        % Count errors
        errors_nccflc =
sum(xor(fix((sign(data_decoded_fwd_ctrl_nccflc)+1)/2),data_fwd_ctrl_nccflc));
        errors_eack =
sum(xor(fix((sign(data_decoded_fwd_ctrl_eack)+1)/2),data_fwd_ctrl_eack));
        for c2 = 1:8
            errors_rcst_ind(c2) =
sum(xor(data_decoded_rtn_ctrl_rcst(:,c2),data_rtn_ctrl_rcst));
        end
        errors_rcst = sum(errors_rcst_ind);
        errors_ecs = sum(xor(data_decoded_rtn_ctrl_ecs,data_rtn_ctrl_ecs));
        station_no = active_station_info(active_index(c));
        biterrors_fwd_ctrl_nccflc(station_no) =
biterrors_fwd_ctrl_nccflc(station_no) + errors_nccflc;

```

```

        biterrors_fwd_ctrl_eack(station_no) =
biterrors_fwd_ctrl_eack(station_no) + errors_eack;
        biterrors_rtn_ctrl_rcst(station_no) =
biterrors_rtn_ctrl_rcst(station_no) + errors_rcst;
        biterrors_rtn_ctrl_ecs(station_no) =
biterrors_rtn_ctrl_ecs(station_no) + errors_ecs;
        if errors_nccflc
            frameerrors_fwd_ctrl_nccflc(station_no) =
frameerrors_fwd_ctrl_nccflc(station_no) + 1;
        end
        if errors_eack
            frameerrors_fwd_ctrl_eack(station_no) =
frameerrors_fwd_ctrl_eack(station_no) + 1;
        end
        if errors_rcst
            frameerrors_rtn_ctrl_rcst(station_no) =
frameerrors_rtn_ctrl_rcst(station_no) + 1;
        end
        if errors_ecs
            frameerrors_rtn_ctrl_ecs(station_no) =
frameerrors_rtn_ctrl_ecs(station_no) + 1;
        end
        if station_no == spec_stat
            biterrors_fwd_ctrl_specific(csec) =
biterrors_fwd_ctrl_specific(csec) + errors_nccflc + errors_eack;
            biterrors_rtn_ctrl_specific(csec) =
biterrors_rtn_ctrl_specific(csec) + errors_rcst + errors_ecs;
        end
    end
end
% TRAFFIC CHANNEL

for c = 1:active_sum
    station_no = active_station_info(active_index(c));
    % Generate protocol frames
    data_fwd_traf = randint(424,1); % random placeholder for forward traffic
    data_rtn_traf = randint(424,1); % random placeholder for return traffic
    if any (ind_ku == c) % Do ku-band channel

        FMT_type(csec,active_index(c)) = 10;
        CNR_fwd_traf_lin = 1/(1/(10^(CNR_fwd_up_ku(csec)/10)) +
1/(10^(CNR_fwd_down_ku(csec,active_index(c))/10)));
        CNR_rtn_traf_lin = 1/(1/(10^(CNR_rtn_up_ku(csec,active_index(c))/10))
+ 1/(10^(CNR_rtn_down_ku(csec)/10)));
        % Encode
        data_coded_fwd_traf = turbocodeDVB(data_fwd_traf,212,'1/2'); % turbo
    encode data
        data_coded_rtn_traf = turbocodeDVB(data_rtn_traf,212,'1/2'); % turbo
    encode data
        % Modulate
        data_coded_int_fwd_traf = bit2intmex(data_coded_fwd_traf,2); % convert
    2 bits to integer
        data_coded_int_rtn_traf = bit2intmex(data_coded_rtn_traf,2); % convert
    2 bits to integer
        data_modulated_fwd_traf =
genqammod(data_coded_int_fwd_traf*ones(1,27),const_QPSK); % Modulate with QPSK
        data_modulated_rtn_traf =
genqammod(data_coded_int_rtn_traf*ones(1,4),const_QPSK); % Modulate with QPSK
        for cfrm = 1:50
            % Add noise to a number of ATM frames according to protocol
            % for each frame
            data_noisy_up_fwd_traf =
awgn(data_modulated_fwd_traf,CNR_fwd_up_ku(csec),'measured');
            data_noisy_up_rtn_traf =
awgn(data_modulated_rtn_traf,CNR_rtn_up_ku(csec,active_index(c)),'measured');
            data_noisy_down_fwd_traf =
awgn(data_noisy_up_fwd_traf,CNR_fwd_down_ku(csec,active_index(c)),'measured');
            data_noisy_down_rtn_traf =
awgn(data_noisy_up_rtn_traf,CNR_rtn_down_ku(csec),'measured');
            % Calculate log-likelihoods & decode
            for c2 = 1:27
                LLR2_fwd_traf =
mod2llr(data_noisy_down_fwd_traf(:,c2),const_QPSK,CNR_fwd_traf_lin,4);
                LLRs_fwd_traf = Somap(LLR2_fwd_traf,0);
                data_decoded_fwd_traf =
turbodecodeDVB(LLRs_fwd_traf,212,'1/2',8,0,data_fwd_traf);

```



```

        errors_fwd_traf =
sum(xor(fix((sign(data_decoded_fwd_traf)+1)/2),data_fwd_traf));
        biterrors_fwd_traf(station_no) =
biterrors_fwd_traf(station_no) + errors_fwd_traf;
        if station_no == spec_stat
            biterrors_fwd_traf_specific(csec) =
biterrors_fwd_traf_specific(csec) + errors_fwd_traf;
        end
    end
    for c2 = 1:4
        LLR2_rtn_traf =
mod2llr(data_noisy_down_rtn_traf(:,c2),const_QPSK,CNR_rtn_traf_lin,4);
        LLRs_rtn_traf = Somap(LLR2_rtn_traf,0);
        data_decoded_rtn_traf =
turbodecodeDVB(LLRs_rtn_traf,212,'1/2',8,0,data_rtn_traf);
        errors_rtn_traf =
sum(xor(fix((sign(data_decoded_rtn_traf)+1)/2),data_rtn_traf));
        biterrors_rtn_traf(station_no) =
biterrors_rtn_traf(station_no) + errors_rtn_traf;
        if station_no == spec_stat
            biterrors_rtn_traf_specific(csec) =
biterrors_rtn_traf_specific(csec) + errors_rtn_traf;
        end
    end
end
else % Do V-band channel
    CNR_fwd_traf_lin = 1/(1/(10^(CNR_fwd_up_V(csec)/10)) +
1/(10^(CNR_fwd_down_V(csec,active_index(c))/10)));
    CNR_rtn_traf_lin = 1/(1/(10^(CNR_rtn_up_V(csec,active_index(c))/10)) +
1/(10^(CNR_rtn_down_V(csec)/10)));
    % Determine FMT scheme
    if all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(1)+scint_margin)
        code_rate = FMT_code_rates{1};
        mod_type = FMT_bits_per_symbol(1);
        FMT_type(csec,active_index(c)) = 1;

        elseif all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(2)+scint_margin)
        code_rate = FMT_code_rates{2};
        mod_type = FMT_bits_per_symbol(2);
        FMT_type(csec,active_index(c)) = 2;

        elseif all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(3)+scint_margin)
        code_rate = FMT_code_rates{3};
        mod_type = FMT_bits_per_symbol(3);
        FMT_type(csec,active_index(c)) = 3;

        elseif all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(4)+scint_margin)
        code_rate = FMT_code_rates{4};
        mod_type = FMT_bits_per_symbol(4);
        FMT_type(csec,active_index(c)) = 4;

        elseif all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(5)+scint_margin)
        code_rate = FMT_code_rates{5};
        mod_type = FMT_bits_per_symbol(5);
        FMT_type(csec,active_index(c)) = 5;

        elseif all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(6)+scint_margin)
        code_rate = FMT_code_rates{6};
        mod_type = FMT_bits_per_symbol(6);
        FMT_type(csec,active_index(c)) = 6;

        elseif all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(7)+scint_margin)
        code_rate = FMT_code_rates{7};
        mod_type = FMT_bits_per_symbol(7);
        FMT_type(csec,active_index(c)) = 7;

        elseif all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(8)+scint_margin)
        code_rate = FMT_code_rates{8};
        mod_type = FMT_bits_per_symbol(8);

```

```

        FMT_type(csec,active_index(c)) = 8;

        elseif all(CNR_fwd_pred_V(csec-3:csec,active_index(c)) >
FMT_CNR_reqs(9)+scint_margin)
            code_rate = FMT_code_rates{9};
            mod_type = FMT_bits_per_symbol(9);
            FMT_type(csec,active_index(c)) = 9;

        else
            if FMT_type(csec-1,active_index(c)) < 10
                code_rate = FMT_code_rates{FMT_type(csec-1,active_index(c))};
                mod_type = FMT_bits_per_symbol(FMT_type(csec-
1,active_index(c)));
                FMT_type(csec,active_index(c)) = FMT_type(csec-
1,active_index(c));
            elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(1)+scint_margin
                code_rate = FMT_code_rates{1};
                mod_type = FMT_bits_per_symbol(1);
                FMT_type(csec,active_index(c)) = 1;

            elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(2)+scint_margin
                code_rate = FMT_code_rates{2};
                mod_type = FMT_bits_per_symbol(2);
                FMT_type(csec,active_index(c)) = 2;

            elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(3)+scint_margin
                code_rate = FMT_code_rates{3};
                mod_type = FMT_bits_per_symbol(3);
                FMT_type(csec,active_index(c)) = 3;

            elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(4)+scint_margin
                code_rate = FMT_code_rates{4};
                mod_type = FMT_bits_per_symbol(4);
                FMT_type(csec,active_index(c)) = 4;

            elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(5)+scint_margin
                code_rate = FMT_code_rates{5};
                mod_type = FMT_bits_per_symbol(5);
                FMT_type(csec,active_index(c)) = 5;

            elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(6)+scint_margin
                code_rate = FMT_code_rates{6};
                mod_type = FMT_bits_per_symbol(6);
                FMT_type(csec,active_index(c)) = 6;

            elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(7)+scint_margin
                code_rate = FMT_code_rates{7};
                mod_type = FMT_bits_per_symbol(7);
                FMT_type(csec,active_index(c)) = 7;

            elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(8)+scint_margin
                code_rate = FMT_code_rates{8};
                mod_type = FMT_bits_per_symbol(8);
                FMT_type(csec,active_index(c)) = 8;

            elseif mean(CNR_fwd_pred_V(csec-3:csec,active_index(c))) >
FMT_CNR_reqs(9)+scint_margin
                code_rate = FMT_code_rates{9};
                mod_type = FMT_bits_per_symbol(9);
                FMT_type(csec,active_index(c)) = 9;

        end

    end

    % Encode
    data_coded_fwd_traf = turbocodeDVB(data_fwd_traf,212,code_rate); %
turbo encode data
    data_coded_rtn_traf = turbocodeDVB(data_rtn_traf,212,code_rate); %
turbo encode data

```

```

        % Modulate
        if mod_type == 2
            data_coded_int_fwd_traf = bit2intmex(data_coded_fwd_traf,2); %
convert 2 bits to integer
            data_coded_int_rtn_traf = bit2intmex(data_coded_rtn_traf,2); %
convert 2 bits to integer
            data_modulated_fwd_traf =
genqammod(data_coded_int_fwd_traf*ones(1,27),const_QPSK); % Modulate with QPSK
            data_modulated_rtn_traf =
genqammod(data_coded_int_rtn_traf*ones(1,4),const_QPSK); % Modulate with QPSK
        else
            data_modulated_fwd_traf =
genqammod(data_coded_fwd_traf*ones(1,27),const_BPSK); % Modulate with BPSK
            data_modulated_rtn_traf =
genqammod(data_coded_rtn_traf*ones(1,4),const_BPSK); % Modulate with BPSK
        end
        for cfrm = 1:50
            % Add noise to a number of ATM frames according to protocol
            % for each frame
            data_noisy_down_fwd_traf =
awgn(data_modulated_fwd_traf,CNR_fwd_V(csec,active_index(c)),'measured');
            data_noisy_down_rtn_traf =
awgn(data_modulated_rtn_traf,CNR_rtn_V(csec),'measured');
            % Calculate log-likelihoods & decode
            for c2 = 1:27
                if mod_type == 2
                    LLR2_fwd_traf =
mod2llr(data_noisy_down_fwd_traf,const_QPSK,CNR_fwd_traf_lin,4);
                else
                    LLR2_fwd_traf =
mod2llr(data_noisy_down_fwd_traf,const_BPSK,CNR_fwd_traf_lin,2);
                end
                LLRs_fwd_traf = Somap(LLR2_fwd_traf,0);
                data_decoded_fwd_traf =
turbodecodeDVB(LLRs_fwd_traf,212,code_rate,8,0,data_fwd_traf);
                errors_fwd_traf =
sum(xor(fix((sign(data_decoded_fwd_traf)+1)/2),data_fwd_traf));
                biterrors_fwd_traf(station_no) =
biterrors_fwd_traf(station_no) + errors_fwd_traf;
                if station_no == spec_stat
                    biterrors_fwd_traf_specific(csec) =
biterrors_fwd_traf_specific(csec) + errors_fwd_traf;
                end
            end
            for c2 = 1:4
                if mod_type == 2
                    LLR2_rtn_traf =
mod2llr(data_noisy_down_rtn_traf,const_QPSK,CNR_rtn_traf_lin,4);
                else
                    LLR2_rtn_traf =
mod2llr(data_noisy_down_rtn_traf,const_BPSK,CNR_rtn_traf_lin,2);
                end
                LLRs_rtn_traf = Somap(LLR2_rtn_traf,0);
                data_decoded_rtn_traf =
turbodecodeDVB(LLRs_rtn_traf,212,code_rate,8,0,data_rtn_traf);
                errors_rtn_traf =
sum(xor(fix((sign(data_decoded_rtn_traf)+1)/2),data_rtn_traf));
                biterrors_rtn_traf(station_no) =
biterrors_rtn_traf(station_no) + errors_rtn_traf;
                if station_no == spec_stat
                    biterrors_rtn_traf_specific(csec) =
biterrors_rtn_traf_specific(csec) + errors_rtn_traf;
                end
            end
        end
    end
end
    end
    disp(['Time elapsed: ' num2str(toc)]);
end
BER_fwd_traf_specific = biterrors_fwd_traf_specific/(50*27*424);
BER_rtn_traf_specific = biterrors_rtn_traf_specific/(50*4*424);
BER_fwd_ctrl_specific = biterrors_fwd_ctrl_specific/(50*(848+96));
BER_rtn_ctrl_specific = biterrors_rtn_ctrl_specific/(50*(8*20+60));
save('Final_detailed_test_results_4station');

```